# MODELO DE NIVELACIÓN DE CARGA DE TRABAJO DE OPERARIOS EN TALLERES CON MÁQUINAS PARALELAS SEMIAUTOMÁTICAS

DAVID ELBERTO RUIZ RUIZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES

BUCARAMANGA

2011

# MODELO DE NIVELACIÓN DE CARGA DE TRABAJO DE OPERARIOS EN TALLERES CON MÁQUINAS PARALELAS SEMIAUTOMÁTICAS

**AUTOR:**

**DAVID ELBERTO RUIZ RUIZ**

**TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE:**

**INGENIERO INDUSTRIAL**

**DIRECTOR:**

**Ingeniero M. Sc. Néstor Raúl Ortíz Pimiento**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS**

**ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES**

**BUCARAMANGA**

**2011**

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS

# RESUMEN

**TÍTULO**

MODELO DE NIVELACIÓN DE CARGA DE TRABAJO DE OPERARIOS EN TALLERES CON MÁQUINAS PARALELAS SEMIAUTOMÁTICAS[*].

**AUTOR**

DAVID ELBERTO RUIZ RUIZ[†].

**PALABRAS CLAVES**

Programación, máquinas paralelas, máquinas semiautomáticas, tiempos de preparación, tiempos setup, carga de trabajo, balanceo de carga.

**DESCRIPCIÓN**

Se resuelve el problema de la balanceo de carga de trabajo en operarios de un sistema productivo formado por máquinas paralelas idénticas semiautomáticas. El objetivo es minimizar los desequilibrios de las cargas de los operarios en las máquinas. Para realizar cierto tipo de trabajo, la máquina tiene un tiempo de preparación dependiente de la secuencia, un tiempo de procesamiento automático y un tiempo semiautomático de carga y descarga (hombre-máquina).
A un operario se le asigna sólo una máquina y cada máquina estará a cargo de un solo operario. Así el número de máquinas equivale al número de operarios. Los trabajos se pueden procesar en cualquier máquina y éstos solo necesitan un solo procesamiento, por ser un proceso intermitente, el reproceso no es válido.
La carga total de una máquina no podrá exceder el tiempo disponible máximo de la jornada de trabajo.
El problema es resuelto mediante un modelo lineal entero para brindar una solución exacta. Aunque en problemas con gran tamaño, el tiempo computacional es muy alto. La capacidad del algoritmo solución con un tiempo razonable es cuantificada. El algoritmo solución es programado con el lenguaje GAMS.
Para cada operador su tiempo de ocio se iguala para mejorar las condiciones laborales como la fatiga por ejemplo.


---

[*] Proyecto de investigación.

[†] Facultad de ingenierías físico-mecánicas. Escuela de estudios industriales y empresariales. Director: Néstor Raúl Ortíz Pimiento.

# ABSTRACT


**TITLE**

AN OPERATOR LOAD BALANCING PROBLEM IN A SEMI AUTOMATIC PARALLEL MACHINE SHOP[‡]

**AUTHOR**

DAVID ELBERTO RUIZ RUIZ[§]

**KEY WORDS**

Scheduling, parallel machines, semi-automatic machines, setup times, load balancig.

**DESCRIPCION**

A load-balancing problem among several operators in a semi-automatic parallel machine shop is resolved. The objective is minimize the unbalance of the workloads among operators. For each job, the machine has a setup sequence-dependent time, Automatic processing time and loading and unloading time semi-automatic (man-machine).

A worker is assigned only one machine and each machine will be headed by a single operator. This way the number of machines equals the number of operators. Jobs can be processed on any machine and they only need a single process, to be an intermittent process, rework is not valid.

The total load of a machine can not exceed maximum time available working day.

The problem is solved by means of a linear entire model to offer an exact solution. Though in problems with great size, the computational time is very high. The capacity of the algorithm solution with a reasonable time is quantified. The algorithm solution is programmed by the language GAMS.

For every operator his time of leisure is equal to improve the working conditions as the fatigue for example.


[‡] Research Project.

[§] Faculty of engineering physicist-mechanics. School of industrial and managerial studies. Director: Néstor Raúl Ortíz Pimiento.

# INTRODUCCIÓN

Para iniciar este proyecto de pregrado se buscó por un problema de programación de operaciones donde se utilizará máquinas semiautomáticas, para facilidad de la programación escogieron máquinas paralelas idénticas por sus características de igualdad en tiempos de procesamiento, y se optó por estudiar la carga de trabajo en general.

La programación de operaciones de máquinas paralelas en sistemas productivos ha tenido diversas metas de estudio como la minimización de costos, minimización de tiempos, maximización de utilidades, minimización de recursos, etc. Un punto de vista desde la perspectiva de la ingeniería humana que surge con la utilización de máquinas semiautomáticas y teniendo como objetivo mejorar la equivalencia de tiempo productivo de cada operario es un aspecto importante dentro de cualquier factoría, ya que tiene bastantes aspectos positivos como el de brindar un mejor ambiente laboral,  mantener un mejor control de la fatiga de los operarios, etc.

Este proyecto de investigación se desarrolla un algoritmo para brindar solución al problema de la nivelación de carga de trabajo en operarios que manipulan máquinas paralelas semiautomáticas y atendiendo recomendaciones de estudios preliminares, los tiempos setup serán dependientes de la secuencia.

## TABLA DE CUMPLIMIENTO DE OBJETIVOS

| OBJETIVO | CUMPLIMIENTO |
|---|---|
| **1. Profundizar la revisión bibliográfica sobre la programación de operaciones, programación de máquinas paralelas, medidas de desempeño y nivelación de carga de trabajo.** | CAPÍTULO 2 Y 3 |
| **2. Definir los parámetros específicos que caracterizarán el proceso de producción.** | CAPÍTULO 4 |
| **3. Definir la función objetivo y las restricciones del modelo de programación lineal.** | CAPÍTULO 4 Y 5 |
| **4. Emplear el lenguaje de modelado GAMS para escribir y posteriormente resolver el problema** | CAPÍTULO 5 |
| | |

# 1. DESCRIPCIÓN DEL PROYECTO

## 1.1. TÍTULO DEL PROYECTO

MODELO DE NIVELACIÓN DE CARGA DE TRABAJO DE OPERARIOS EN TALLERES CON MÁQUINAS PARALELAS SEMIAUTOMÁTICAS.

## 1.2. OBJETIVOS

### 1.2.1. OBJETIVO GENERAL

- Proponer un modelo de programación lineal para resolver el problema de balance de carga de trabajo de operarios que tienen a su cargo máquinas paralelas idénticas semiautomáticas con tiempos de alistamiento (tiempos setup) dependiente de la secuencia.

### 1.2.2. OBJETIVOS ESPECIFICOS

- Profundizar la revisión bibliográfica sobre la programación de operaciones, programación de máquinas paralelas, medidas de desempeño y nivelación de carga de trabajo.
- Definir los parámetros específicos que caracterizarán el proceso de producción.
- Definir la función objetivo y las restricciones del modelo de programación lineal.
- Emplear el lenguaje de modelado GAMS para escribir y posteriormente resolver el problema

## 1.3. ALCANCE DEL TRABAJO DE GRADO

Dentro del desarrollo de este proyecto se involucrará el modelamiento matemático del problema, así como la programación lineal utilizando el lenguaje GAMS® para programar y buscar solución al modelo matemático.

De acuerdo con lo anterior, los resultados y productos a entregar en el presente trabajo de investigación son:

- Documento que incluya el marco teórico de la programación de máquinas paralelas, con sus respectivas revisiones bibliográficas y el algoritmo de solución al problema de nivelación de carga de trabajo para máquinas paralelas semiautomáticas idénticas con setup dependiente de la secuencia.
- Modelo escrito en GAMS®.
- Artículo académico.

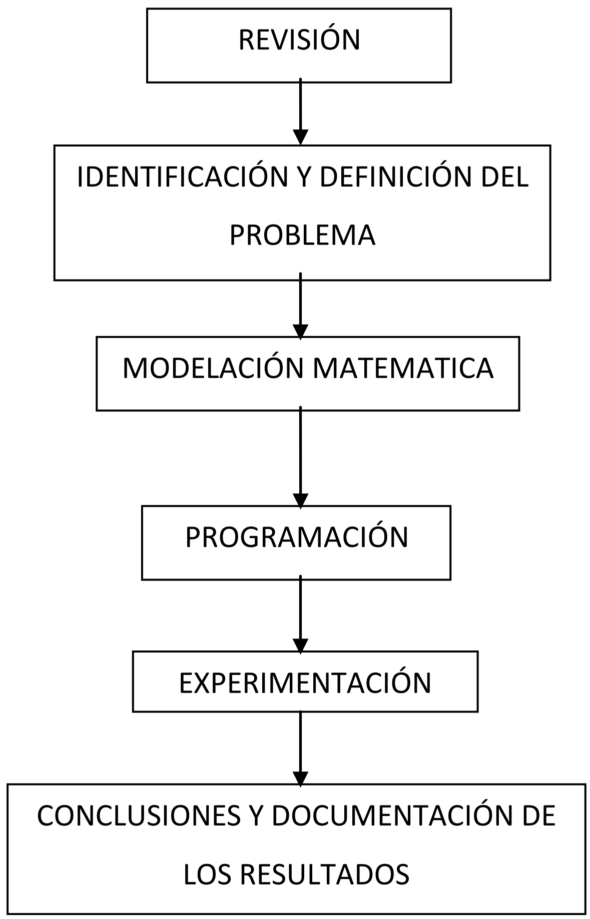
## 1.4. METODOLOGIÁ DE LA INVESTIGACIÓN

La metodología utilizada en esta investigación aparece en la Figura 1. Una breve descripción de los pasos que se siguieron en el desarrollo de la investigación.

*Consulta Bibliográfica*: Búsqueda y profundización en textos, libros, documentos académicos y bases de datos sobre: la programación de máquinas paralelas, nivelación de carga de trabajo en operarios y/o máquinas, medidas de desempeño en la programación de operaciones y conceptos básicos. Así como también, la documentación y organización de temas de los últimos adelantos que se han hecho referente a la nivelación de carga (máquinas y operarios) y programación de máquinas paralelas.

*Definición del problema*: Dentro de la consulta bibliográfica y en concertación con los últimos documentos académicos se define el problema en el cual se pretende dar una solución, teniendo en cuenta el interés del grupo de investigación, el alcance, requerimiento del nivel de proyecto de grado (pregrado, especialización, maestría), así como también la configuración del problema de la programación de operaciones. Comprendiendo las diferentes variaciones que contiene la programación de máquinas paralelas (tiempos con o sin setup, máquinas paralelas idénticas, proporcionales o distintas, máquinas automáticas o semiautomáticas) y estudiar los diferentes desarrollos heurísticos y exactos en la solución de estos problemas, así como la observación de su metodología para el desarrollo de los diferentes problemas.

Figura 1 Metodología de la investigación




*Fuente: Autor*

*Modelación matemática*: De acuerdo con la definición del problema se propone la modelación matemática del problema con su función objetivo y las restricciones correspondientes, teniendo en cuenta aspectos como horas disponibles en la jornada de trabajo, recursos disponibles, etc.

*Programación*: Se tendrá el lenguaje GAMS® como primer lenguaje de programación para crear el algoritmo solución y resolver el problema matemático, sin embargo se utilizará el lenguaje MATLAB®, WinQSB y/o el solver de EXCEL® para corroborar datos obtenidos en el primer lenguaje.

*Experimentación*: Se propondrá ejercicios teóricos para darles solución con el algoritmo solución que será programado con el lenguaje GAMS®. Se presentará un cuadro con los diferentes problemas numéricos.

*Conclusiones y documentación de los resultados*: Con los resultados obtenidos durante la experimentación del algoritmo se logrará dar conclusiones acerca del estudio y mostrar diferentes investigaciones que podrán ser desarrolladas a la conclusión del proyecto.

# 2. MARCO TEÓRICO CONCEPTUAL

## 2.1. INVESTIGACIÓN DE OPERACIONES

La investigación de operación determina el mejor camino de un problema de decisión con una serie de restricciones, está asociado a la rama de las matemáticas que usa la estadística, algoritmos y modelos matemáticos para el proceso de toma de decisiones.

### 2.1.1. ¿Qué es la Investigación de Operaciones?

Según G. Durán [1] una definición que se acerca mucho a la realidad sería "la ciencia de la toma de decisiones", es la aplicación del método científico a problemas relacionados con el control de las organizaciones o sistemas, a fin de que se produzcan soluciones que sirvan a los objetivos de la organización.

Algo muy cerca de lo que años atrás definieron CHURCHMAN, ACKOFF Y ARNOFF: "La investigación de operaciones es la aplicación, por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (Hombre-Máquina), a fin de que se produzcan soluciones que mejor aproveche los objetivos de la organización".

LA SOCIEDAD AMERICANA DE SISTEMAS DE PRODUCCIÓN Y CONTROL DE INVENTARIOS, (APICS) La define como el análisis cualitativo de operaciones industriales y administrativas con el intento de derivar un entendimiento integrado de los factores que controlan los sistemas operacionales en vista de proporcionar a la administración un objeto básico para tomar decisiones que frecuentemente involucran representar por un medio matemático la realidad.

### 2.1.2. Objetivos

- La construcción de modelos de decisión basados en descripciones matemáticas, con el objetivo de tomar decisiones en situaciones de complejidad o incertidumbre.

- Simular eventos con el fin de pronosticar problemas o dificultades
- Controlar sistemas y/o organizaciones
- Conocer el riesgo de la toma de decisiones
- Brindar soluciones en miras al objeto de la organización.

### 2.1.3. Métodos

La investigación de operaciones incluye un conjunto muy amplio de técnicas orientadas a proporcionar una ayuda cuantitativa a la toma de decisiones, el método empleado es el método científico, y las técnicas que se utilizan son, en buena medida, técnicas matemáticas.

Algunos técnicas/métodos utilizados en la solución de problemas son:

- Programación lineal, entera, dinámica, no lineal, multiobjetivo.
- Programación matemática
- Modelos de Transporte
- Modelos de Redes
- Programación estocástica
- Gestión de inventarios
- Teoría de Juegos
- Simulación
- Fenómenos de espera (colas)
- Búsqueda Tabú
- Algoritmo de recorrido simulado
- Las redes neuronales
- Algoritmos genéticos
- Metaheurísticas
- Heurísticas

Técnicas/Métodos que plantean soluciones a problemas específicos como: Planeación de la Producción, Asignación de personal, Transporte, Inventarios, Mercado, Estrategias de Inversión, Etc.

## 2.2. Programación Lineal

Según R. Chase [2]  la programación lineal se refiere a varias técnicas matemáticas utilizadas, en forma óptima, los recursos limitados a distintas demandas que compiten por ellos. La Programación Lineal es la técnica matemática más popular para la optimización y se ha aplicado a muchos problemas de la administración de operaciones, algunas de las aplicaciones típicas son:

- Planeación de operaciones y ventas agregadas.
- Planeación de productos.
- Rutas de Productos.
- Programación de vehículos/cuadrillas.
- Control de procesos.
- Estudios para ubicar la planta.
- Manejo de materiales.
- Etc.

Para que una situación del área de dirección de operaciones planteé un problema de programación lineal debe cumplir con cinco condiciones básicas:

a) Recursos limitados: cantidad de suministros, equipamiento, dinero y/o materiales.
b) Objetivo específico: como maximizar la ganancia o minimizar el costo.
c) Linearidad: los parámetros obedecen a la linearidad entre ellos, es decir si necesita dos horas para realizar una pieza, entonces dos piezas tomarían cuatro horas.
d) Homogeneidad: todas las horas en que trabaja el operario son igual de productivas.
e) Divisibilidad: la programación lineal asume que todos los parámetros, productos y recursos se pueden subdividir en fracciones.

Cuando la subdivisión no es posible dado a las características del recurso (como la contratación de 1/3 de persona), se  puede utilizar una modificación de la programación lineal llamada Programación Entera.

Cuando el objetivo es único (maximizar, minimizar), se puede utilizar la programación lineal. Cuando existen varios objetivos, se utiliza la programación por metas. Si el problema se resuelve por etapas o plazos de tiempos, se utiliza la programación dinámica. Cuando se tiene restricciones de otra naturaleza tal vez requiera que se resuelva con variantes de estas técnicas, como la programación no lineal y/o programación cuadrática.

El problema de la programación lineal entraña un proceso de optimización en el cual se eligen valores no negativos para una serie de variables $X_1, X_2, \ldots, X_n$ de modo que se maximice (o minimice) una función objetivo:

Maximizar (minimizar) $Z = C_1 X_1 + C_2 X_2 + \ldots + C_n X_n$

Sujeto a las restricciones de los recursos con la fórmula:

$A_{11} X_1 + A_{12} X_2 + \ldots + A_{1n} X_n \leq B_1$

$A_{21} X_1 + A_{22} X_2 + \ldots + A_{2n} X_n \leq B_2$

.

.

.

$A_{m1} X_1 + A_{m2} X_2 + \ldots + A_{mn} X_n \leq B_m$

Donde $C_n$, $A_{mn}$, $B_m$ son parámetros dados.

Dependiendo del problema, las restricciones se pueden expresar con signo de igualdad (=) o con signo de mayor/menor o igual que ($\geq$, $\leq$).

## 2.3. CLASIFICACIÓN DE LA MÁQUINA EN UNA LÍNEA DE PRODUCCIÓN

### 2.3.1. Máquinas Paralelas

Según I. Vila et al al [3] las máquinas paralelas hacen referencia a máquinas similares que están en la capacidad de procesar el mismo trabajo y se clasifican en idénticas, proporcionales y distintas, descritas a continuación:

#### 2.3.1.1. Máquinas paralelas idénticas

Dos máquinas son idénticas si para todo trabajo el tiempo de fabricación es el mismo en cualquiera de ellas.

#### 2.3.1.2. Máquinas paralelas uniformes o proporcionales

Son proporcionales si algunas de las máquinas son capaces de procesar los trabajos más rápido (o lento) que las otras.

#### 2.3.1.3. Máquinas paralelas diferentes o distintas

Son distintas si el tiempo de proceso varía en función de la máquina utilizada y del trabajo asignado, es el caso general de las anteriores (idénticas y uniformes). También son llamadas máquinas paralelas no relacionadas.

De la definición anterior se pueden ampliar las siguientes palabras.

### 2.3.2. Clasificación de las máquinas según su tipo

Según I. Vila et al [4] las máquinas pueden ser automáticas, manuales o semiautomáticas:

#### 2.3.2.1. Máquinas automáticas:

Se consideran máquinas automáticas a aquellas máquinas que trabajan sin necesitar la continua presencia y control de las personas. A través de un programa se indica a la máquina la operación que se desea, como ordenadores, máquinas CNC, etc.

#### 2.3.2.2. Máquinas manuales:

Se consideran máquinas manuales a aquellas máquinas que trabajan con la necesidad de la continua ´presencia y control de personas. Estas máquinas no poseen ningún tipo de motor y se accionan sus mecanismos al recibir fuerzas externas.

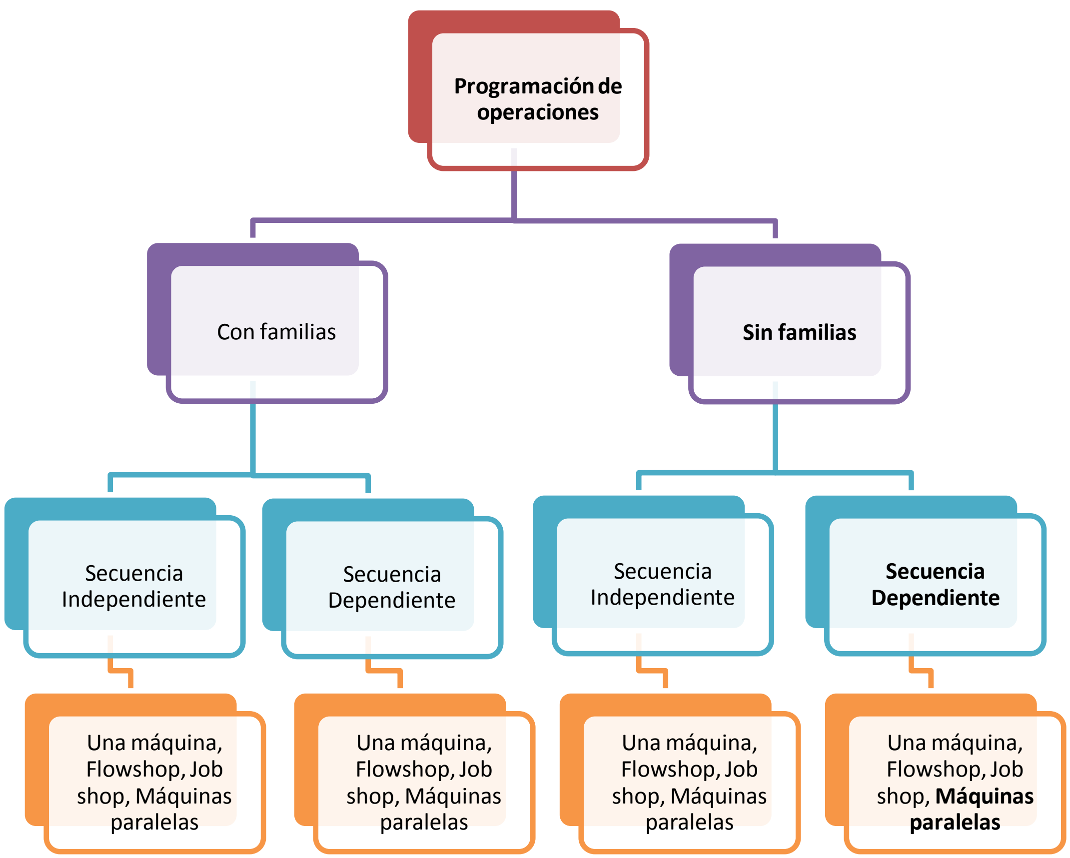
#### 2.3.2.3. Máquinas semiautomáticas:

Se considera máquinas semiautomáticas a aquellas máquinas que poseen características de máquinas manuales y automáticas teniendo un tiempo de maquinado conjunto (hombre-máquina) y tiempo de procesamiento automático para cumplir a cabalidad el ciclo de la máquina.

## 3. ESTADO DEL ARTE

Según D'Armas [5] la programación de operaciones ha sido un tema estudiado durante los últimos 40 años. Las investigaciones se agrupan en cuatro ambientes productivos principales: una máquina, flowshop, job shop y máquinas paralelas, y se clasifican en problemas con y sin familias, con tiempos de preparación dependientes e independientes de la secuencia, como se observa en el siguiente cuadro conceptual:

Figura 2 Clasificación de la programación de operaciones




*Fuente: revista UNIVERSIDAD, CIENCIA Y TECNOLOGÍA, Volumen 10, Número 34, junio 2005, pág. 101*

Según Fanjul [6] la investigación de operaciones inició sus estudios con seriedad en la industria con los trabajos de Henry Gantt [7], [8] y otros pioneros que siguieron sus ideas. Pasaron varias décadas para que aparecieran las primeras publicaciones en la literatura de la investigación de operaciones, artículos como el de Smith [9], Johnson [10] y Jackson [11], trabajos que enfatizaron en la programación tanto entera como dinámica para la programación de la producción. En la década de 1970 y 1980 la investigación se centro en la complejidad de problemas de la programación de producción, siguiendo diferentes direcciones tanto en la parte teórica como en la parte práctica.

Una de las diferentes direcciones fueron las investigaciones que se enfocaron con máquinas paralelas y sus clasificaciones (idénticas, uniformes y distintas), así, para esta investigación se presenta los antecedentes teóricos respecto al tema de interés de máquinas paralelas idénticas.

Considerando el problema de máquinas paralelas idénticas, las cuales tienen la capacidad de procesar cualquier trabajo en cualquier máquina en el mismo tiempo, el estudio de Graham et al. [12] aborda este problema teniendo como objetivo la minimización del makespan, con lo cual Garey y Johnson [13] demostraron que este problema pertenecía al tipo NP-hard. Muchos algoritmos de aproximación y heurísticos fueron desarrollados, experimentando un tiempo computacional alto. Como concluyo Sahni [14] al desarrollo de unos algoritmos más complicados que podían obtener el resultado tan cercanos al óptimo como se desea, pero a cambio de un alto crecimiento exponencial del tiempo de ejecución. Consecutivamente, Shmoys [15] modificó los algoritmos para obtener una mejora en el tiempo computacional. Dell' Amico y Martello [16] consiguieron mejorar todos los algoritmos hasta la fecha. Posteriormente Mokotoff [17] presentó un trabajo basado en Ramificación y Acotación, con lo cual un año más tarde Dell' Amico y Martello [18] respondieron con un trabajo también basado en Ramificación y Acotación con excelentes resultados. Finalmente Dell' Amico et al.

[19] propusieron dos nuevos algoritmos, uno exacto y otro metaheurístico basado en un ‘scaner search’ con buenos resultados.

A continuación se describe algunos aportes encontrados en la literatura científica, que vale la pena destacar con relación a la programación de operaciones con máquinas paralelas en general.

Kravchenko et al [20] investigaron sobre la secuenciación de operaciones en máquinas paralelas, en la que un servidor realiza la preparación de las máquinas, con el objetivo de minimizar el tiempo de procesamiento. Por lo cual Lee et al [21] agrego una regla dispatching y un procedimiento de Recorrido Simulado para el problema de máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia.

Liaee et al [22] consideraron la programación por familias de operaciones con tiempos de preparación dependientes en una máquina o en máquinas paralelas, con o sin la suposición de tecnología de grupo que las operaciones en cada familia deben ser programadas continuamente.

Brucker [23] presenta una serie de casos especiales de asignación de tareas: piezas con restricciones de precedencia; piezas con fechas de entrega; tiempos de preparación; y tiempos de transporte. Utilizando una clase de algoritmo heurístico conocido como método de descomposición planea la secuenciación y asignación de tareas a gran escala para sistemas de fabricación complejos.

Balakrishnan et al [24] consideraron el problema de programar n operaciones en m máquinas paralelas que operan a velocidades diferentes (conocido como máquinas paralelas uniformes), para minimizar la suma de los costos de los adelantos y de los retrasos. Presentaron un modelo matemático de formulación entera mixta. Por lo cual Sivrikaya-Serifoglu et al [25] incluyeron en la formulación del problema suposiciones como distintas fechas de vencimientos, distintos tiempos de llegada de las operaciones, diferentes tasas de procesamiento para las máquinas y tiempos de preparación dependientes de la secuencia, esto para darle un enfoqué más real al problema, y emplearon dos propuestas de algoritmos genéticos para abordar el problema.

Park et al [26] trataron el problema de programación con máquinas paralelas con un conjunto de operaciones esperando en una cola, los tiempos de preparación son dependientes de la secuencia y el objetivo fue minimizar el retraso total.

Bilge et al [27] estudiaron el problema de un conjunto de piezas independientes, con preparaciones dependientes de la secuencia y emplearon una aproximación búsqueda tabú para atacar este problema.

F. Charrua Santos et al [28] describen un problema de programación de tareas en máquinas paralelas uniformes, con los tiempos setup dependientes de la secuencia. El problema que se analiza contiene un determinado conjunto de restricciones, incluida la capacidad de equipos, precedencias de trabajo, tamaño de lote y el plan de entrega de tareas. La complejidad del modelo de programación matemática desarrollado en este problema, no ha permitido encontrar una solución mediante la optimización de los métodos exactos, razón por la cual los autores han desarrollado una metaheurística basada en el recocido simulado[**], algoritmo que permite obtener soluciones casi óptimas. Utilizaron el algoritmo de recocido IMULATED, para buscar la solución del modelo. Así, los autores trataron de demostrar en este estudio enfocado básicamente a la programación de máquinas paralelas en la producción de textiles que, tal cual esta propuesto el modelo matemático es difícil encontrar una solución con un método

[**] *Recocido Simulado: El método fue descrito independientemente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983.*

exacto, luego con la utilización de esta metaheurística permite encontrar una solución cercana a la óptima.

Farouk Yalaoui et al [29] Los autores describen el problema de optimización multiobjetivo que se centra en la programación de maquinas paralelas con n trabajos que se deben asignar a m máquinas idénticas en paralelo con fechas de lanzamiento, fechas de vencimiento, donde el objetivo es minimizar dos objetivos (makespan y la tardanza total). En este artículo plantean un modelo exacto y dos modelos aproximados (NSGA-II[††] y SPEA-II[‡‡]). La configuración de tiempos setup depende de la secuencia sij, el tiempo de preparación cambia cuando la producción pasa del trabajo i al trabajo j (el trabajo i precede al trabajo j en la misma máquina). Cada máquina puede ejecutar sólo un trabajo a la vez y cada trabajo sólo se puede procesarse una vez.
Los autores resuelven el problema a través de un modelo exacto y comparan estos resultados con los obtenidos por dos métodos aproximados (ya mencionado con anterioridad). El método exacto encuentra las soluciones óptima paretianas. La aplicación de los dos algoritmos aproximados escogidos por los autores (NSGA-II y SPEA-II) aparte de mostrar su estructura, muestran la calidad de los algoritmos en los resultados computacionales, los experimentos numéricos muestran para casos pequeños como para casos grandes, se resalta la NSGA-II, la cual en pequeños casos brinda una solución óptima en un tiempo razonable (mucho menor tiempo computacional que SPEA-II), y muestra que para más de 8 trabajos el algoritmo no arroja resultados.

Muchas investigaciones han desarrollado métodos para minimizar costos, tiempos y/o recursos dentro de la programación de operaciones de máquinas paralelas,

---

[††] *NSGA-II: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization - K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan (2000)*
[‡‡] *SPEA2: improving the strength pareto evolutionary algorithm - E. Zitzler, M. Laumanns, and L. Thiele (2001)*

dejando de un lado una parte importante dentro de la ingeniería humana como lo es el equilibrio de la carga de trabajo en operarios, cuando se trabaja con máquinas semiautomáticas, el cual ha tenido poco avances en el ámbito científico. Unos de los artículos que proporcionan un estudio donde involucra la carga de trabajo de los operarios es el de Dug Hee Moon et al [30] que aborda el problema de equilibrio de carga de trabajo con operarios de máquinas paralelas idénticas semiautomáticas con setup independientes de la secuencia, en las que existe dos tipos de máquina que pueden ser operadas, cada máquina sólo puede hacer un tipo de trabajo, sin embargo cada trabajador está en la capacidad de operar ambos tipos de máquina, los trabajos se reúnen en dos grupos (G1, G2).

Trabajos G1 a procesar en las N máquinas de tipo 1
Trabajos G2 a procesar en las M máquinas de tipo 2

En ningún caso los trabajos podrán pasar por la maquina tipo 1 y consecutivamente pasar por el tipo 2 (no es tipo flowshop), tampoco los trabajos G1 pueden ser procesados en la máquina tipo 2, ni los trabajos G2 pueden ser procesados en la máquina tipo 1. Así el objetivo es asignar los trabajos a las máquinas adecuadas y asignar las máquinas a los operarios cuando la función objetivo es minimizar el desequilibrio de la carga de trabajo entre los operadores.
El problema que formulan los autores es un programa lineal entero y dan una solución jerárquica que se compone de cuatro fases, en la primera fase los trabajos se asignan a los operadores que utilizando para ésto un algoritmo genético cuyo objetivo es minimizar el desequilibrio de la carga total del operador, sin tener en cuenta las posibles limitaciones (horas máquina disponible, horas jornada de trabajo, etc.). La segunda fase, se determina cuantas máquinas tipo 1 y tipo 2 deben ser asignadas a cada operador para satisfacer la carga de trabajo calculada a partir de los resultados de la fase 1. La tercera fase es un generador de soluciones factibles, donde se encuentran las máquinas que están violando las restricciones del problema, por lo tanto son modificadas intercambiando máquinas

y variando el orden, todo esto hasta cumplir con todas las restricciones del problema y quedar con una solución factible y así pasar a la cuarta fase que por medio de un algoritmo proceden a mejorar la solución factible obtenida en la tercera fase.

Durante del recorrido del estado del arte de programación de máquinas paralelas se evidencia la carencia de modelos para la nivelación de carga en operarios, así la definición del problema de investigación de este proyecto de grado, va a suplir tal necesidad, el desarrollo estará enfocado específicamente en máquinas paralelas con tiempos setup dependientes de la secuencia (acogiendo unas de las recomendaciones propuestas por Dug Hee Moon), para mayor detalle nuestro próximo capítulo estará dedicado a la descripción con más énfasis del problema a resolver.

# 4. DESCRIPCIÓN Y DISEÑO DEL PROBLEMA A RESOLVER

## 4.1. DESCRIPCIÓN DEL PROBLEMA

EL objeto de esta investigación es el equilibrio de la carga de trabajo de operarios que manipulan máquinas semiautomáticas. Así se tendrá una mayor equidad entre los operarios de determinada factoría. La definición respecto a los tiempos setups (tiempos de alistamiento) se evidencia en la necesidad al mostrar un enfoque realista al problema ya que se resalta la importancia del tiempo de alistamiento de la máquina pues en la mayoría de casos depende de la secuencia del trabajo que se va a procesar.

El problema clásico de máquinas paralelas ha sido ampliamente tratado en la literatura debido a la cantidad de aplicaciones que tiene en los procesos de manufactura reales. Casi la totalidad de los aportes investigativos en este tema han tratado el problema considerando únicamente la carga de trabajo asignada a las máquinas que componen el sistema de producción en paralelo. Sin embargo, es poco común analizar la carga de trabajo de los operadores de dicha maquinaria, sobretodo, cuando las máquinas son semiautomáticas, es decir, no son manuales (en donde la carga de trabajo de la máquina corresponde exactamente a la carga de trabajo del operario asignado a dicha máquina) o automáticas (en donde no hay presencia de operarios).

El análisis del trabajo con máquinas semiautomáticas es un tema clásico abordado desde la disciplina de ingeniería de métodos, en donde se busca asignar y nivelar la carga de trabajo de los operarios cuando trabajan con máquinas semiautomáticas. Tradicionalmente se resuelve este problema de manera heurística, por medio de gráficas conocidas como los diagramas hombre máquina (operator-machine chart) y/o los diagramas de actividades múltiples (multiple activity chart).

Combinando los problemas anteriores (“Parallel machine problem” y “operator load-balancing problem”) se obtiene un primer problema sencillo conocido como “operator load-balancing problem in a semi-automatic parallel machine shop”, resuelto inicialmente por Dug Hee Moon et al [30] para secuenciar Trabajos (J) con tiempos de alistamiento (Setup times) independientes de la secuencia de procesamiento de dichos trabajos (J).

El presente trabajo de investigación se propone abordar el problema: “operator load-balancing problem in a semi-automatic parallel machine shop”, cuando los tiempos de alistamiento (Setup times) dependen de la secuencia de procesamiento de los Trabajos (J).

### 4.1.1. Hipótesis Fundamental

El problema de balance de la carga de trabajo de operarios que laboran en un taller con máquinas semiautomáticas idénticas y en paralelo, con tiempos de alistamiento dependientes de la secuencia de procesamiento de los trabajos, puede ser resuelto por medio de un modelo de programación lineal entera mixta.

### 4.1.2. Delimitación del problema

Se trabajará para el caso de un Proceso Intermitente en máquinas paralelas semiautomáticas. De acuerdo con Dominguez Machuca [31] se clasificaría como Configuración productiva por lotes.

## 4.2. DISEÑO DEL MODELO MATEMÁTICO

Basado en la descripción del problema se representará a continuación un modelo Lineal Entero, el cual estará enfocado en la nivelación de la carga de trabajo de operarios que manipulan máquinas semiautomáticas paralelas idénticas, pero antes de todo se definen algunos supuestos necesarios para llevar a cabalidad el modelo matemático.

### 4.2.1. Supuestos

- Máquinas semiautomáticas.
- Máquinas paralelas idénticas (cada máquina tiene la capacidad de procesar cualquier trabajo en un tiempo igual).
- Un tipo de máquina.
- M máquinas paralelas idénticas en el taller.
- A un operario se le asignará sólo una máquina y cada máquina estará a cargo de un solo operario. Así el número de máquinas equivale al número de operarios.
- Cada trabajo podrá tener una determinada cantidad de unidades.
- Cada unidad de producto requiere una sola operación en una sola de las máquinas paralelas.
- Cada trabajo tiene un tiempo de maquinado conjunto hombre-máquina (Se refleja en la realidad al montaje y desmontaje de un producto en determinada máquina) y un tiempo automático (sólo máquina)
- Tiempos setup dependiente de la secuencia de los trabajos.
- Los tiempos setup se consideran tiempo de maquinado conjunto (hombre-máquina).
- En ningún caso el tiempo total de la carga de cada máquina superará al tiempo disponible de trabajo.

### 4.2.2. Datos de Entrada

- Número de máquinas.
- Número de trabajos.
- Unidades de producto que contiene un trabajo específico.
- Matriz tiempos setup.
- Tiempos de trabajo conjunto hombre-máquina
- Tiempo de procesamiento automático.
- Tiempo máximo disponible de la jornada de trabajo.

### 4.2.3. Notaciones

$H$ Tiempo máximo disponible de la jornada de trabajo

$m$ Número de máquinas k.

$n$ Número de trabajos i.

$P_i$ Tiempo de maquinado automático de la unidad del producto tipo i.

$L_i$ Tiempo de maquinado conjunto (hombre-máquina) de una unidad del trabajo i.

$Q_i$ Unidades del producto que contienen el trabajo i.

$S_{ij}$ Tiempo setup del trabajo i que precede al trabajo j.

$W_K$ Carga total de un operario en la máquina k.

$AW$ Promedio de carga de trabajo de un operario en una máquina.

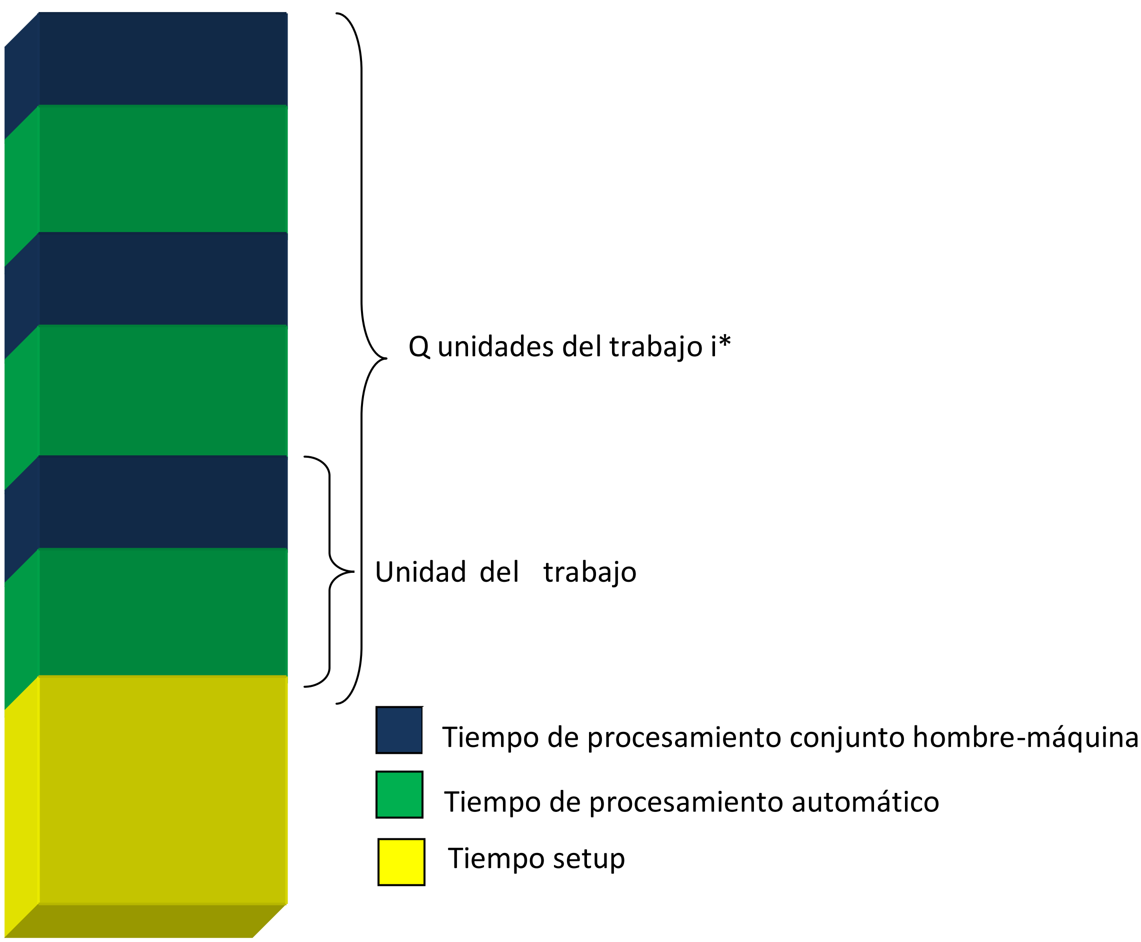
$X_{ijk}$ 1, si el trabajo i precede al trabajo j en la máquina k, 0 en otro caso.

### 4.2.4. Características del problema matemático

En la Figura 3 se muestra un esquema gráfico que permite entender la forma que está organizado el conjunto de tiempos que conforma un trabajo i, así como la carga de la máquina k.

Los tiempos setup se hacen al final de cada trabajo, Se asume que la máquina se deja lista para el trabajo que sigue. Luego para la programación se crea 2 variables ficticias binarias ($X_{ijk}$), por lo que el primer trabajo sólo tendrá el tiempo de setup partiendo desde la tarea cero hacia la tarea j ($S_{0j}$), acompañado de sus tiempos respectivos $P_0$ y $L_0$ que tendrán un valor igual a cero ($P_0 = 0$ y $L_0 = 0$). Con la variable ficticia que representa la finalización del trabajo y su ubicación en la parte final de la secuencia de trabajo en la respectiva máquina k, su tiempo setup será igual a cero ($S_{i,\ n+1} = 0$), puesto que lógicamente después de este trabajo no se programara más trabajos para esta máquina.

Figura 3 Representación del bloque de trabajo.




**Para el bloque de trabajo se muestran 3 unidades del producto que conforman el trabajo i.*


*** Para el bloque de trabajo en la máquina k se muestran 2 trabajos que conforman la carga de la máquina.*

*Fuente: Autor*

El objetivo de esta investigación es minimizar el nivel de carga de cada operario, la representación matemática seria:

$$Min = \sum_{K=1}^{m} (AW - \mathrm{W_k})$$

Como se trata de una diferencia surge el problema con los valores negativos y positivos, con el fin de darle un tratamiento adecuado y con el objetivo de lograr la programación de la función objetivo se investiga como los autores resuelven un problema de maximización/minimización de una diferencia de dos variables como por ejemplo el artículo de Dug Hee Moon et al [30] muestra el cálculo de una diferencia cuadrática, donde la diferencia de estas dos variables se encuentra elevada al cuadrado, así resuelve el problema de diferencias negativas.

$$Min = \sum_{K=1}^{m} (AW - \mathrm{W_k})^2$$

El software GAMS ofrece la función valor absoluto “abs (x)”, sin embargo el autor de este proyecto en desarrollo desconoce la forma del método numérico interno del software para el cálculo de esta función.

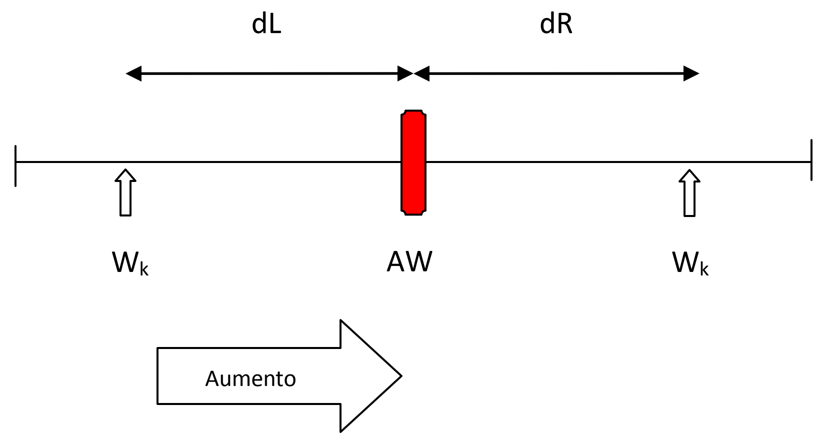
$$Min = \sum_{K=1}^{m} abs\ (AW - \mathrm{W_k})$$

Como el objetivo de esta investigación es el desarrollo de un modelo lineal, Smith et al [32] propone un método de sustitución de variables para reemplazar el valor absoluto de una diferencia, este método crea nuevas variables de sustitución y deja al problema del valor absoluto en términos de las variables transformadas y operaciones aritméticas simples. Para esta investigación se efectúa esta última opción de la diferencia del cálculo de la carga de trabajo. A continuación se expone el método de sustitución e integrando a la adaptación al problema de interés de esta investigación:

Función objetivo a sustituir:

$$Min = \sum_{K=1}^{m} |\ AW - \mathrm{W_k}\ |$$

Luego el valor $W_k$ podrá tener un valor menor, igual o mayor que el valor promedio AW.




Si algún $W_k$ excede la meta $\Rightarrow$ dR > 0 y dL = 0

Si algún $W_k$ no llega a la meta $\Rightarrow$ dL > 0 y dR = 0

Si algún $W_k$ cumple la meta $\Rightarrow$ dR = 0 y dL = 0

Luego para cada operario:

$dR = \frac{1}{2} ( |AW - W_k| - (AW - W_k) )$

$dL = \frac{1}{2} ( |AW - W_k| + (AW - W_k) )$

Entonces:

$dR_k + dL_k = |AW - W_k|$

$dL_k - dR_k = AW - W_k$

$dL_k - dR_k + W_k = AW$ (nueva restricción para cada máquina k)

Nueva función objetivo $Min = \sum_{K=1}^{m} ( dR_k + dL_k )$

Se trata de crear dos variables llamadas en este caso $d^-$ y $d^+$ que indican los acercamientos por la izquierda y la derecha respectivamente, así como el cambio de la función objetivo y la adicción de una nueva restricción.

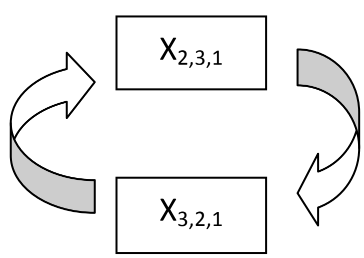
Por otra parte, en los problemas de secuenciación de trabajos y problemas de ruteo de vehículos existe un problema que toma la solución final, se trata de unos “subciclos” (también llamados en la literatura *subtours*) que toma internamente la solución.

Para ejemplificar esta situación, se analizara a continuación un problema con 2 máquinas y 8 trabajos, y se presentara como solución que la primera máquina realice los 5 primeros trabajos, la segunda máquina los 3 trabajos restantes.

Máquina 1 : $X_{0,1,1} - X_{1,4,1} - X_{4,5,1} - X_{5,9,1} - X_{2,3,1} - X_{3,2,1}$

Máquina 2: $X_{0,6,2} - X_{6,7,2} - X_{7,8,2} - X_{8,9,2}$

Observando la secuenciación de la máquina 1, se denota la iniciación del trabajo 1, en secuencia con el trabajo 4 y finalmente con el trabajo 5; los trabajos 3 y 2 se presenta el estado de subciclos, y significaría físicamente que del trabajo 2 se transfiera al trabajo 3, y que luego nuevamente el trabajo 3 se transfiera al trabajo 2.




Aunque el algoritmo presenta una respuesta numérica válida, no es aceptable esta respuesta pues se trata de obtener una respuesta totalmente secuencial. No solamente los subciclos están compuestos de 2 trabajos, esto sucede para 3 y 4 trabajos, y depende de la cantidad de trabajos, entre más trabajos se asigne a una máquina más grande podrá ser el subciclo. Así como también pueden existir uno o más subciclos dentro de la misma programación de trabajos de la máquina respectiva.

En artículo de Alfredo Olivera [33] sobre ruteo de vehículos, propone una restricción fácilmente programable en GAMS que resuelve el problema de los subciclos y que es integrada al modelo propuesto en este trabajo de investigación.

En esta nueva restricción del modelo, aparece la variable U, la cual, es una variable auxiliar para eliminar subciclos en la maquina k. El subíndice i es el trabajo que precede, j es el trabajo que le sigue inmediatamente, N es el número de trabajos. $X_{ijk}$ es 1 si el trabajo i precede inmediatamente al trabajo j en la maquina k.

$$\mathrm{U_{ik}} - \mathrm{U_{jk}} + \mathrm{N} \cdot \mathrm{X_{ijk}} \leq \mathrm{N} - 1 \qquad \forall\ i, j, k \qquad i \in \{1 \ldots n\}$$

$$j \in \{1 \ldots n+1\}$$

$$k \in \{1 \ldots m\}$$

Finalmente, surge la necesidad de restringir la secuencias que parten desde un trabajo y programan el mismo trabajo, es decir trabajos que no tienen secuencia (como por ejemplo $X_{1,1,k}$ o $X_{3,3,k}$) aunque se puede programar en iniciación de variables iguales a cero (en la parte superior del software GAMS) se opta por hacer un vector y volverlo restricción formal del algoritmo dado que para un alto número de trabajos la representación del vector se vuelve extensa al escribir cada uno de sus posibles combinaciones.

Luego,

$$\mathrm{X_{ijk}} = 0 \qquad \forall\ i = j \qquad i \in \{1 \ldots n\}$$

$$j \in \{1 \ldots n+1\}$$

Una vez terminada estos últimos tres aspectos importantes para la programación del algoritmo solución se presenta en nuestro siguiente subtitulo la formulación formal del algoritmo solución final.

### 4.2.5. Función Objetivo

$$Min = \sum_{K=1}^{m} ( DR_k + DL_k)$$

### 4.2.6. Restricciones

| | | |
|---|---|---|
| $\sum_{j=1}^{n} X_{0jk} = 1$ | ∀ k  k ∈ {1…m} | (1) |
| $\sum_{K=1}^{m} \sum_{i=0}^{n} X_{ijk} = 1$ | ∀ j  j ∈ {1…n+1} | (2) |
| $\sum_{K=1}^{m} \sum_{j=1}^{n+1} X_{ijk} = 1$ | ∀ i  i ∈ {1…n} | (3) |
| $\sum_{i=0}^{n} X_{ijk} = \sum_{i=1}^{n+1} X_{jik}$ | ∀ j,k  j ∈ {1…n+1}  k ∈ {1…m} | (4) |
| $\sum_{i=1}^{n} X_{in+1k} = 1$ | ∀ k  k ∈ {1…m} | (5) |
| $\sum_{i=0}^{n} \sum_{j=1}^{n+1} X_{ijk} ( S_{ij} + Q_i(L_i + P_i)) \leq H$ | ∀ k  k ∈ {1…m} | (6) |
| $\sum_{i=0}^{n} \sum_{j=1}^{n+1} X_{ijk} ( S_{ij} + Q_i(L_i)) = W_k$ | ∀ k  k ∈ {1…m} | (7) |
| $AW = (1/m) \sum_{k=1}^{m} W_k$ | ∀ k  k ∈ {1…m} | (8) |
| $U_{ik} - U_{jk} + N \cdot X_{ijk} \leq N - 1$ | ∀ i, j ,k<br>i ∈ {1…n}  j ∈ {1…n+1}  k ∈ {1…m} | (9) |
| $DL_k - DR_k + W_k = AW$ | ∀ k  k ∈ {1…m} | (10) |
| $X_{ijk} = 0$ | ∀ i = j  i ∈ {1…n}  j ∈ {1…n+1} | (11) |
| $X_{ijk} \in \{0,1\}$ | | |

La ecuación (1) asigna a cada máquina un trabajo de iniciación “$X_{0jk}$”, cuyos tiempos de procesamiento son iguales a cero y sólo van a tener tiempos setups. Las ecuaciones (2) y (3) aseguran que cada trabajo sólo es asignado a una máquina. La ecuación (4) denota la correcta secuenciación de los trabajos. La ecuación (5) hace referencia que cada máquina hace un trabajo ficticio final “n+1”, "$X_{i,n+1,k}$", cuyos tiempos setups son iguales a cero y sólo tendrá tiempos de procesamiento que indican la realización de la tarea i. Ecuación (6) describe que la carga total de una máquina no puede exceder el tiempo disponible. Ecuación (7) expresa la carga total de cada trabajador en su máquina. La ecuación (8) indica la carga promedio de trabajo de un operario. Ecuación (9) evita los subciclos en el sistema. Ecuación (10) es la conversión de variables para reemplazar el valor absoluto y la Ecuación (11) establece la exclusión de variables sin secuencia.

# 5. PROGRAMACIÓN Y EXPERIMENTACION NUMÉRICA DEL ALGORITMO SOLUCIÓN

Para la programación y desarrollo del algoritmo se utilizó el software GAMS® (sistema general de modelado algebraico por sus siglas en ingles), GAMS® es un software especializado que proporciona a través de su lenguaje de sistemas de modelado de alto nivel para la programación matemática y optimización.

Con el fin de mostrar el uso del algoritmo solución, así como la entrada de datos al código se propone el siguiente ejercicio numérico:

Número de máquinas = 2 Máquinas

Número de trabajos = 9 trabajos

Cantidad de unidades de cada trabajo:

$Q_1$ = 10 unidades

$Q_2$ = 10 unidades

$Q_3$ = 10 unidades

$Q_4$ = 10 unidades

$Q_5$ = 10 unidades

$Q_6$ = 10 unidades

$Q_7$ = 10 unidades

$Q_8$ = 10 unidades

$Q_9$ = 10 unidades

Tiempos de procesamiento conjunto hombre-máquina (carga-descarga):

$L_1$ = 3 Segundos

$L_2$ = 4 Segundos

$L_3$ = 6 Segundos

$L_4$ = 4 Segundos

$L_5$ = 4 Segundos

$L_6$ = 5 Segundos

$L_7$ = 6 Segundos

$L_8$ = 11 Segundos

$L_9$ = 9 Segundos

Tiempos de procesamiento automático:

$P_1$ = 15 Segundos

$P_2$ = 13 Segundos

$P_3$ = 19 Segundos

$P_4$ = 17 Segundos

$P_5$ = 11 Segundos

$P_6$ = 13 Segundos

$P_7$ = 15 Segundos

$P_8$ = 14 Segundos

$P_9$ = 19 Segundos

Tiempos setup (Tiempos de alistamiento de las máquinas):

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 12 | 10 | 5 | 11 | 12 | 8 | 9 | 12 | 5 |
| 1 | - | 12 | 3 | 9 | 7 | 5 | 6 | 4 | 7 |
| 2 | 10 | - | 8 | 8 | 18 | 2 | 4 | 8 | 4 |
| 3 | 8 | 9 | - | 10 | 8 | 9 | 5 | 11 | 13 |
| 4 | 6 | 7 | 4 | - | 5 | 9 | 2 | 11 | 7 |
| 5 | 7 | 5 | 4 | 7 | - | 6 | 5 | 6 | 1 |
| 6 | 11 | 7 | 8 | 6 | 4 | - | 9 | 5 | 7 |
| 7 | 5 | 7 | 8 | 5 | 7 | 8 | - | 6 | 4 |
| 84 | 5 | 8 | 6 | 2 | 5 | 11 | 16 | - | 7 |
| 9 | 11 | 9 | 3 | 6 | 7 | 8 | 4 | 5 | - |

*unidades de tiempo en Segundos

Se presenta el código de GAMS, así como la introducción de datos:

```
* salida por cada grupo de restricciones igual a 100 restricciones
option limrow = 100;
*salida por cada grupo de variables igual a 100 variables
option limcol = 100;
SETS
*El trabajo 0 y 10 son ficticios

         II       Trabajos                        /0, 1*10/

         I(ii)    Trabajos                        /1*10/

         T(i)     TRABAJOS REALES final j         /1*9/

         M        Maquina                         /1,2/

         N        Numero de trabajos              /1*9/
```

```
PARAMETER

         Q(ii)  Cantidad de unidades del trabajo i
         /0 = 0
          1 = 10
          2 = 10
          3 = 10
          4 = 10
          5 = 10
          6 = 10
          7 = 10
          8 = 10
          9 = 10
          10 = 0/

         L(ii)  Tiempo de carga y descarga del trabajo i
         /0 = 0
          1 = 3
          2 = 4
          3 = 6
          4 = 4
          5 = 4
          6 = 5
          7 = 6
          8 = 11
          9 = 9
          10 = 0/


         P(ii)  Tiempo de procesamiento del trabajo n
         /0 = 0
          1 = 15
          2 = 13
          3 = 19
          4 = 17
          5 = 11
          6 = 13
          7 = 15
          8 = 14
          9 = 19
          10 = 0/


*DISPLAY II, M, N, TP, Q, L, P;


SCALARS

         K       Numero de maquinas iguales      / 2 /
         H       Tiempo disponible (minutos)    /999/
```

```
Table  S(ii,ii)    Tiempo de alistamiento

       0       1       2       3       4       5       6       7       8       9       10

0      999     12      10      5       11      12      8       9       12      5       1
1      0       999     12      3       9       7       5       6       4       7       0
2      0       10      999     8       8       18      2       4       8       4       0
3      0       8       9       999     10      8       9       5       11      13      0
4      0       6       7       4       999     5       9       2       11      7       0
5      0       7       5       4       7       999     6       5       6       1       0
6      0       11      7       8       6       4       999     9       5       7       0
7      0       5       7       8       5       7       8       999     6       4       0
8      0       5       8       6       2       5       11      16      999     7       0
9      0       11      9       8       6       7       8       4       5       999     0

VARIABLE

                 FT             FUNCION DE TIEMPO EQUILIBRAR

POSITIVE VARIABLES

        TM             TIEMPO DEL CICLO
        AW             PROMEDIO DE CARGA DE TRABAJO DEL OPERARIO
        W(M)           CARGA DE TRABAJO DEL OPERARIO EN LA MAQUINA K
        DL(M)          Distancia Izquierda
        DR(M)          Distancia Derecha
        U(II,M)        ELIMINACION DE SUBCICLOS


BINARY VARIABLES
                 X(ii,ii,M)      Tarea i que precede a j en la maquina m;

*........................................................................
* En la siguiente inicializacion, se coloca 0 cuando i=j (que se opto por programarlo
como restricción)

X.FX("0","10","1")= 0;
X.FX("0","10","2")= 0;


EQUATIONS

   Z               DIFERENCIA DE TIEMPO A MINIMIZAR

   R1(T)           RESTRICCIÓN DE ASIGNACION UNICA PARA EL TRABAJO i

   R2(M)           RESTRICCION QUE INICIA CADA MAQUINA TIENE UNA TAREA DE INICIACION
FICTICIA (0)

   R3(T,M)            RESTRICCION QUE INDICA LA SECUENCIACIÓN DE LOS TRABAJOS Y DEBEN
PROCESARSE EN LA MISMA MAQUINA

   R4(M)           RESTRICCION QUE INDICA QUE LA CARGA DE UNA MAQUINA NO PUEDE EXCEDER EL
TIEMPO DISPONIBLE
```

```
R5(M)            RESTRICCION QUE INDICA LA CARGA DEL OPERARIO EN CADA MAQUINA

R6               RESTRICCION QUE INDICA LA CARGA DE TRABAJO PROMEDIO

R7(M)            RESTRICCION QUE REEMPLAZA EL VALOR ABSOLUTO  DL-Left DR-Right

R8(T)            RESTRICCIÓN QUE INDICA QUE SE REALIZZCION DEL TRABAJO i

R9(M)            RESTRICCION QUE INDICA QUE CADA MAQUINA HACE UN TRABAJO FICTISIO FINAL

R10(I,M)         INICIZACION DE VARIABLES = 0

R11(T,I,II,M)    ELIMINACION DE SUBCICLOS;


Z..               FT =E= SUM(M,DL(M)+DR(M));


R1(T)..           SUM((II,M), X(II-1,T,M)) =E= 1 ;
R2(M)..           SUM(T, X("0",T,M)) =E= 1  ;
R3(T,M)..         SUM(II, X(II-1,T,M)) =E= SUM(I, X(T,I,M));
R4(M)..      SUM((II,I),(Q(ii-1)*L(ii-1)+Q(ii-1)*P(ii-1)+S(ii-1,I))*X(ii-1,I,M)) =L= H;
R5(M)..           SUM((II,I),(Q(ii-1)*L(ii-1)+S(ii-1,I))*X(ii-1,I,M)) =E= W(M);
R6..              (1/k)*SUM(M,W(M)) =E= AW;
R7(M)..           W(M)-DR(M)+DL(M) =E= AW;
R8(T)..           SUM((I,M), X(T,I,M)) =E= 1 ;
R9(M)..           SUM(T, X(T,"10",M)) =E= 1  ;
R10(I,M)..        X(I,I,M) =E= 0;
R11(T,I,II,M)..  U(T,M)-U(I,M)+9*X(T,I,M) =L= 8;


MODEL XXX /ALL/;
Option Iterlim = 1E8;
Option Reslim  = 1E8;
Option Optcr   = 0.0001
SOLVE XXX MINIMIZING FT using MIP;
```

RESULTADOS:

Resultado de la función objetivo

$FT = 0$

Secuenciación de los trabajos en cada máquina:

Máquina 1: $X_{0\,8\,1}$ , $X_{8\,4\,1}$ , $X_{4\,1\,1}$ , $X_{1\,5\,1}$ , $X_{5\,2\,1}$ , $X_{2\,10\,1}$

Máquina 2: $X_{0\,7\,2}$ , $X_{7\,6\,2}$ , $X_{6\,9\,2}$ , $X_{9\,3\,2}$ , $X_{3\,10\,2}$

Carga promedio de trabajo de los operarios:

$AW = 292$ unidades de tiempo

Carga de trabajo del operario en la máquina k:

$W_{\text{máquina 1}} = 292$ unidades de tiempo

$W_{\text{máquina 2}} = 292$ unidades de tiempo

Con el ánimo de mostrar algunos de los ejercicios numéricos realizados en la experimentación numérica se presenta a continuación las siguientes variaciones de problemas:

| Número de Máquinas | Número de Trabajos | Tiempo de Corrida | No. de Restricciones | Solución de la Función Objetivo |
|---|---|---|---|---|
| 2 | 4 | 1 [seg] | 187 | 0 |
| 2 | 8 | 1 [seg] | 1.509 | 0 |
| 2 | 9 | 3 [seg] | 2.055 | 0 |
| 3 | 9 | 182 [seg] | 3.068 | 0 |
| 3 | 16 | 523 [seg] | 14.832 | 0 |
| 3 | 20 | 681 [seg] | 28.808 | 0 |
| 3 | 40 | 822 [seg] | 207.016 | 0 |
| 4 | 16 | 45 [min] | 19.775 | 0 |
| 4 | 20 | 192 [min] | 37.191 | 0 |
| 4 | 40 | t > 6[horas] | 275.951 | - |
| 6 | 20 | t > 4[horas] | 55.451 | - |
| 6 | 24 | t > 6[horas] | 93.977 | - |

Tiempos de corrida se ejecuta en estos experimentos en un procesador AMD Athlom™ X2 Dual-Core 2.17 Ghz con sistema operativo Windows Seven a 32 bits.

# 6. CONCLUSIONES

El tamaño del experimento numérico fue escogido por su facilidad para representar los datos de entrada en este texto, sin embargo este algoritmo ha sido probado hasta con 25 ejercicios numéricos los cuales se destacaron 12 en el capitulo anterior.

En cuanto a la capacidad, se denota un ejercicio conformado con 20 trabajos en 4 máquinas donde en aproximadamente 192 minutos (3 horas y 12 minutos) de corrida arroja una solución, para ejercicios mayores, el tiempo computacional se extiende significativamente, esto por recorrer todas las posibles combinaciones que tienen un aumento polinomial, con el fin de cuantificar dicha afirmación, para un problema de 24 trabajos en 6 máquinas se dejó correr por tiempo de 360 minutos (6 horas) donde el algoritmo todavía trabajaba con sus interacciones.

Se presentó la evolución de la programación del algoritmo hasta llegar al desarrollo por completo del problema de interés, así para estudiantes que no hayan tenido un contacto anterior con la formulación matemática y el lenguaje GAMS, irá a ser mucho más fácil su entendimiento.

Al término de la presente investigación se ha presentado un algoritmo para dar solución al problema de balanceo de carga de operarios que manipulan máquinas paralelas idénticas semiautomáticas, el objetivo es programar trabajos a las máquinas paralelas con el fin de que cada operario tenga similar tiempo productivo, sin importar su tiempo de ocio.

Se abre una nueva línea de investigación en el grupo de investigación OPALO (Grupo de Optimización y Organización de Sistemas Productivos, Administrativos y Logísticos de la escuela de estudios industriales y empresariales - UIS) y se entusiasma a seguir investigando sobre los diferentes aspectos que pueden adicionar o trabajar en paralelo a este trabajo. Por lo cual en el siguiente título de las recomendaciones se describen algunos.

# 7. RECOMENDACIONES

De la experimentación numérica se evidencio que para problemas mayores a 24 trabajos el tiempo computacional es muy alto, luego sería muy conveniente desarrollar este problema con algoritmos metaheurísticos para abreviar el tiempo de respuesta.

En futuras investigaciones, se prevé investigaciones multiobjetivo que además de dar el balance de la carga del operario, minimice otras funciones objetivo como por ejemplo el makespan.

En cuanto a estudios posteriores con máquinas paralelas semiautomáticas es de gran interés avanzar el desarrollo de los métodos para dar solución a problemas que tiene que ver con asignación a dos o más máquinas a un operario, así el tiempo de ocio del operario será mejor aprovechado y la solución tomará una respuesta más productiva.

# 8. BIBLIBOGRAFÍA


[1] Guillermo Durán. (2006). Investigación de operaciones, modelos matemáticos y optimización, Centro de Gestión de Operaciones, Departamento de Ingeniería Industrial, Universidad de Chile, Chile. p 314.

[2] Richard Chase, Robert Jacons y Nicholas Aquilano. (2009). Administración de operaciones, Producción y cadena de suministros, Duodécima edición, McGrAW-HILL. p 777.

[3] Imma Ribas Vila y Ramon Companys Pascual. (2006). Programación de la Producción en un sistema flow shop híbrido sin esperas y tiempos de preparación dependientes de la secuencia, Departament d'Organ ització d'Empreses. Escola Tècnica Superior d'Enginyers Industrials de Barcelona, Barcelona. p 38-40.

[4] Imma Ribas Vila y Ramon Companys Pascual. (2006). Programación de la Producción en un sistema flow shop híbrido sin esperas y tiempos de preparación dependientes de la secuencia, Departament d'Organ ització d'Empreses. Escola Tècnica Superior d'Enginyers Industrials de Barcelona, Barcelona. p 41-42.

[5] Mayra D'Armas Regnault. (2005). Universidad, Ciencia y Tecnología, Volumen 10, No. 34: p 96-102.

[6] Luis Fanjul Peyró. (2010). Nuevos algoritmos para el problema de secuenciación en máquinas paralelas no relacionadas y generalizadas. Universidad Politécnica de Valencia. Valencia. p 15-35.

[7] Gantt, H. (1910). Work, Wages and Profit. Their influence on the cost of living. The Engineering Magazine Co, New York.

[8] Gantt, H. (1919). Organizing for Work. Harcourt, Brace and Howe, New York.

[9] Smith, W. E. (1956). Various optimizers for single-stage production. Naval Research Logistics Quarterly, 3(1): p 59–66.

[10] Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1(1): p 61–68.

[11] Jackson, J. R. (1956). An extension of johnson's results on job lot scheduling. Naval Research Logistics Quarterly, 3: p 201–203.

[12] Graham, R. L., Lawler, E. L., Lenstra, J. K., y Rinnooy Kan, A. H. G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. Annals of Discrete Mathematics, 5(1): p 287–326.

[13] Garey, M. R. y Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman. San Francisco.

[14] Sahni, S. (1976). Algorithms for scheduling independent tasks. Journal of the ACM, 23(1): p 116–127.

[15] Shmoys, D. B. y Tardos, E. (1987). An approximation algorithm for the generalized assignment problem. Mathematical Programming, 62(3): p 461–474.

[16] Dell'Amico, M. y Martello, S. (1995). Optimal scheduling of tasks on identical parallel processors. ORSA Journal on Computing, 7(2): p 191–200.

[17] Mokotoff, E. (2004). An exact algorithm for the identical parallel machine scheduling problem. European Journal of Operational Research, 152: p 758–769.

[18] Dell'Amico, M. y Martello, S. (2005). A note on exact algorithms for the identical parallel machine scheduling problem. European Journal of Operational Research, 160: p 576–578.

[19] Dell' Amico, M., Iori, M., Martello, S., y Monaci, M. (2008). Heuristic and exact algorithms for the identical parallel machine scheduling problem. INFORMS Journal on Computing, 20(3): p 333–344.

[20] Kravchenko S y Werner F. (1997). Parallel Machine Scheduling Problems with a Single Server. Mathematical Computer Modelling; 26: p 1-11.

[21] Lee Y y Pinedo M. (1997). Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times. European Journal of Operational Research; 100: p 464-474.

[22] Liaee M y Emmons H. (1997). Scheduling Families of Jobs with Setup Times. International Journal of Production Economics; 51: p 165-176.

[23] Brucker P. (1998). Scheduling algorithms, 2nd edition, Springer, Heidelberg. p 54-69.

[24] Balakrishnan N, Chalet J y Sridharan V. (1999). Early/Tardy Scheduling with Sequence Dependent Setups on Uniform Parallel Machines. Computers & Operations Research No. 26. p 122-141.

[25] Sivrikaya-Serifoglu F y Ulusoy G. (1999). Parallel Machine Scheduling with Earliness and Tardiness Penalties. Computers & Operations Research; 26: p 773–787.

[26] Park Y. Kim S. Lee Y-H Sheduling. (2000). Job son Parallel Machines Applying Neutral Networks and Heuristic Rules. Computers and Industrial Engineering; 38: p 189-202.

[27] Bilge U, Kraç F, Kurtulan M y Pekgün P. (2004). A Tabu Search Algorithm for Parallel Machine Total Tardiness Problem. Computers & Operations Research. Computers & Operations Research No 31. p 172-197.

[28] F. Charrua Santos & Pedro M. Vilarinho. (2010). The Problem of Scheduling in Parallel Machines: A Case Study, Proceedings of the World Congress on Engineering 2010 Vol III, WCE 2010, London, U.K. 13 pages.

[29] Farouk Yalaoui, Xiaohui Li, Lionel Amodeo, & Hicham Chehade. (2010). “A Multiobjective Optimization Approach to Solve a Parallel Machines Scheduling Problem”, Advances in Artificial Intelligence, Volume 2010, Article ID 943050, 10 pages.

[30] Dug Hee Moon, Dae Kyoung Kim, Jong Yun Jung. (2004). “An operator load-balancing problem in a semi-automatic parallel machine shop”, Computers & Industrial Engineering 46; p 355–362.

[31] José A. Domínguez Machuca. (1995). Dirección de operaciones. Aspectos estratégicos en la producción y los servicios, Mc Graw Hill. p 457.

[32] Ricardo Smith Q. Oscar Mesa S. Isaac Dyner R. (2000). “Decisiones con múltiples objetivos e incertidumbre” editorial Universidad Nacional de Colombia, Universidad Nacional de Colombia. Capítulo 17, p 208-210.

[33] Alfredo Olivera. (2004). “Heurística para Problemas de Ruteo de Vehículos”, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay. p 63.