UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

TESIS

PROBLEMA DE ASIGNACIÓN DE TIEMPOS EN LA ORGANIZACIÓN DE UNA LÍNEA DE ENSAMBLAJE

POR

BEATRIZ ALEJANDRA GARCÍA RAMOS

EN OPCIÓN AL GRADO DE MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS

SEPTIEMBRE, 2019

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO

TESIS

PROBLEMA DE ASIGNACIÓN DE TIEMPOS EN LA ORGANIZACIÓN DE UNA LÍNEA DE ENSAMBLAJE

POR

BEATRIZ ALEJANDRA GARCÍA RAMOS

EN OPCIÓN AL GRADO DE MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS

SEPTIEMBRE, 2019

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirector de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Problema de asignación de tiempos en la organización de una línea de ensamblaje», realizada por el alumno Beatriz Alejandra García Ramos, con número de matrícula 1550385, sea aceptada para su defensa como requisito parcial para obtener el grado de Maestría en Ciencias en Ingeniería de Sistemas.

El Comité de Tesis

Dra. Yasmín Águeda Ríos Solís
Directora

Dr. Roger Z. Ríos Mercado
Revisor

Dr. Mário A. Saucedo Espinosa
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Posgrado

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO
U.A.N.L.
FIME

San Nicolás de los Garza, Nuevo León, septiembre 2019

A mis padres Beatriz y Jesús
que me han apoyado siempre a cumplir mis metas.

A mi hermana Karina
que también ha sido mi amiga.

A mi novio Gerardo
que ha vivido conmigo los momentos más difíciles.

A mi familia y amigos
que son mi apoyo incondicional.

Índice general

Agradecimientos ix

Resumen xii

1. Introducción 1

1.1. Hipótesis 2

1.2. Objetivo 3

1.3. Estructura de la Tesis 3

2. Antecedentes y Revisión Bibliográfica 5

2.1. Problema de Producto-Pieza-Molde-Máquina 5

2.2. Modelo Matemático del Problema Producto-Pieza-Molde-Máquina . . 7

2.3. Línea de ensamblaje 11

3. Problema de Línea de Ensamblaje 15

3.1. Descripción del Problema de Línea de Ensamblaje 17

3.2. Algoritmo Base 21

3.3. Ejemplo del Problema de Línea de Ensamblaje 22

4. Algoritmos para el Problema de Línea de Ensamblaje 28

4.1. Algoritmos para el Problema de Línea de Ensamblaje 28

4.1.1. Algoritmo de Ensamblajes Totales 29

4.1.2. Algoritmo de Ensambles Totales y Sub-ensambles 31

4.2. Algoritmo de Procesamiento de Información del Problema Producto-Pieza-Molde-Máquina 37

5. Resultados Experimentales 42

5.1. Periodos p 43

5.2. Productos Ensamblados 45

5.3. Piezas Almacenadas 48

6. Conclusiones y Trabajo a Futuro 53

6.1. Conclusiones 54

6.2. Trabajo a futuro 56

Índice de figuras

2.1. Asignación producto-pieza-molde-máquina [15]. 6

2.2. Amontonamiento de piezas en una línea de ensamblaje. 13

3.1. Respuesta gráfica del Problema Producto-Pieza-Molde-Máquina. . . . 16

3.2. Respuesta gráfica del Problema Producto-Pieza-Molde-Máquina seccionado en periodos p considerados para el Problema de Línea de Ensamblaje. 19

5.1. Resultados de los periodos promedio de los algoritmos 2 y 3. 46

5.2. Resultados de los productos ensamblados de los algoritmos 2 y 3. . . 49

5.3. Resultados de las piezas en almacenamiento de los algoritmos 2 y 3. . 52

Índice de tablas

3.1. Tipos de pieza y cantidad de piezas necesarias para ensamblar los productos. 24

3.2. Respuesta del Problema Producto-Pieza-Molde-Máquina donde se muestra el tipo de pieza j y la cantidad de ese tipo de pieza que surge de la línea de producción en un periodo p. 25

3.3. Respuesta de la línea de ensamblaje para ensambles totales. 26

3.4. Respuesta de la línea de ensamblaje para ensambles totales y subensambles de productos. 27

Agradecimientos

Agradezco principalmente a Dios que día a día me demuestra su infinito amor, que me ha dado el don de la vida y que está para mí en los momentos más difíciles y en los más felices. Agradezco que me permita ver lo hermoso de la vida y lo bello que es tener a personas que me amen a mi lado.

Agradezco a mis padres Jesús García Gámez y Beatriz Ramos Larralde que me han dado una gran vida. Gracias por darme lo más valioso que tienen: sus consejos, su sabiduría, su educación y su amor. Gracias por ser mi ejemplo y mi soporte, por mostrarme siempre que Dios está en nuestra familia, y por nunca darse por vencidos para que Kary y yo tengamos la mejor educación y seamos unas personas de bien; todo lo que han hecho por ambas lo recordaré por siempre y no terminaré de agradecérselos. Si no fuera por ustedes yo no habría llegado a ser quien soy, se merecen el reconocimiento de todos mis logros. Los amo a los dos.

Agradezco a mi hermana Karina Guadalupe García Ramos, quien fue mi apoyo incondicional cuando existieron días difíciles. A ti te debo mis momentos de tranquilidad al conversar cuando llegaba a casa agotada de un día de arduo trabajo de investigación. Gracias por ser también mi amiga. Te amo.

Agradezco a mi novio Francisco Gerardo Meza Fierro, quien durante estos dos años ha comprendido perfectamente lo que es estar en una maestría de este nivel y me ha dado todo su apoyo incondicional como compañero, amigo y novio. Gracias por ser mi pañuelo de lágrimas, mi opuesto perfecto y mi más grande motivador. Gracias por tu tiempo, tu cariño y por todas las ocasiones en que me apoyaste para

seguir adelante cuando algo no iba bien con mi investigación. Gracias por haberme enseñado que cualquier cosa puede superarse si estamos juntos. Te amo.

Agradezco a toda mi familia por estar al pendiente de mi formación académica; a mis tíos, primos y mi abuela, que piensan en mi futuro y me dan consejos sobre mi futuro. Lamento no poder verlos tanto como antes, esto es un trabajo duro que necesita más tiempo del que creí, prometo comunicarme con ustedes más seguido, los quiero demasiado.

Agradezco a mis amigos y compañeros del posgrado que me dieron grandes consejos para seguir adelante y no dejarme vencer por las malas experiencias al inicio de mi investigación. Gracias por escucharme siempre y por esas reuniones en las que me han permitido conocerlos y ser parte de sus vidas, los considero como mi segunda familia.

Agradezco en especial a Víctor Domínguez que ha escuchado mis quejas, mis derrotas, mis frustraciones, mis alegrías y mis logros; gracias por ser un gran amigo y por alegrar mis días en Yalma. Agradezco también a Alberto y Delavy, porque nos han aceptado como parte de su vida a todos, son unas personas increíbles. Agradezco a Alan, Mayra, Gabriela, Yessica, David y Astrid, que más que compañeros de trabajo han sido para mí grandes personas que me han apoyado en algún momento de mi estancia en el posgrado tanto académicamente como personalmente. No tienen idea de cuánto los quiero y aprecio. Agradezco también a Citlali, Pablo, Eduardo y Daniel que gracias a su experiencia en el posgrado nos han apoyado a todos a seguir adelante.

Agradezco a mis amigos y compañeros de la Iglesia que siempre se preocupan por mí y me preguntan sobre mi maestría; a ellos que también han sido mi segunda familia desde hace años: Arturo, Airy, Obed, Paola, sí se pudo.

Agradezco a la Facultad de Ingeniería Mecánica y Eléctrica (FIME) y a la Universidad Autónoma de Nuevo León (UANL) por ofrecer un gran programa académico en el Posgrado en Ciencias en Ingeniería de Sistemas en el cual pude concluir una

etapa académica que me ayudará a superarme a mí misma y por la beca que se me proporcionó para llevar a cabo mis estudios. Agradezco también las instalaciones que proporcionan a nuestro posgrado para que podamos tener un laboratorio en dónde trabajar y salones donde podamos tomar las materias que se nos imparten.

Agradezco a los Doctores que se encuentran en el Posgrado en Ciencias en Ingeniería en Sistemas por ser parte de mi crecimiento académico y por darme sus enseñanzas sobre lo que un investigador y un docente puede llegar a ser. En particular agradezco a los miembros de mi comité de tesis, el Dr. Roger Z. Ríos Mercado y el Dr. Mario A. Saucedo Espinosa, por haber leído mi tesis y haberme apoyado en las correcciones de la misma. De manera especial agradezco a mi asesora Yasmín A. Ríos Solís que durante toda mi investigación estuvo apoyándome y dirigiendo la misma.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca de manutención que me fue otorgada con la cual pude facilitarme mis estudios al utilizar ese ingreso en transporte, comida e incluso en la investigación que durante dos años estuve realizando.

Resumen

Beatriz Alejandra García Ramos.

Candidato para obtener el grado de Maestría en Ciencias en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: Problema de asignación de tiempos en la organización de una línea de ensamblaje.

Número de páginas: 61.

Objetivos y método de estudio: El objetivo de esta investigación es estudiar el problema de la asignación de tiempos en la organización de una línea de ensamblaje e implementar algoritmos que logren simular el comportamiento del sistema que existe en una línea de ensamblaje.

La metodología de solución que se propone consiste en un algoritmo que determina la organización en la línea de ensamblaje de acuerdo a la información que se obtiene de la línea de producción que se analizó en el Problema Producto-Pieza-Molde-Máquina. Esta organización consiste en determinar el periodo en que un producto debe comenzar a ensamblarse e indica la cantidad de piezas que deben ser ensambladas durante ese periodo determinado.

El método de estudio es el análisis de un modelo matemático existente pro-

puesto para la solución de una línea de producción, el cual es la base para la creación de algoritmos heurísticos que resuelven el problema de la línea de ensamblaje.

CONTRIBUCIONES Y CONCLUSIONES: Las principales contribuciones de la investigación son dos. La primera es procesar la información que surge de la línea de producción y realizar con ello instancias que se requieren para la línea de ensamblaje. Esto se logró realizar en base a las instancias que requiere el Problema Producto-Pieza-Molde-Máquina y la respuesta que surge de él, lo cual fue satisfactorio.

La segunda contribución es el diseño y experimentación de los algoritmos que determinan el trabajo que se debe llevar a cabo en la línea de ensamblaje. Estos algoritmos determinan el trabajo de ensamblaje de acuerdo a las características que se le proporcionaron a la línea; uno de los algoritmos es el que logra mejorar el proceso al obtener una gran cantidad de productos finales y lograr que no existan piezas que queden en el almacenamiento.

Firma de la directora: _____

Dra. Yasmín Águeda Ríos Solís

Capítulo 1

Introducción

Una línea de ensamblaje busca principalmente minimizar el personal que es necesario para realizar su trabajo o minimizar el tiempo de ocio que existe entre un ensamble y otro. Para poder llevar a cabo estos objetivos se deben tomar en consideración las estaciones de trabajo que existen para ensamblar los productos, las cuales comúnmente tienen un mecanismo automático para transportar las piezas que deben ser ensambladas y/o los productos sub-ensamblados. Además se debe considerar que el proceso para ensamblar las piezas puede ser realizado por máquinas o por operadores humanos [2].

Para que la línea de ensamblaje pueda lograr su objetivo, se debe tener un balanceo factible en la línea, el cual consiste en asignar cada uno de los productos o de las piezas a una de las estaciones de trabajo de acuerdo a las restricciones que puedan tener éstas en el problema que se presente [5].

Estudios más a fondo de la línea de ensamblaje consideran importante la secuencia que debe de tener el ensamblado de los productos ya que existen productos diversos, y el tiempo que existe entre la salida de una estación y la entrada de otra del producto que se esté ensamblando [29]. Además se debe considerar que existe un tiempo de ciclo, el cual puede establecerse con el tiempo máximo de las estaciones de ensamblaje [30], y que todas las estaciones están equipadas de manera balanceada

con respecto a las máquinas que hay en funcionamiento y los trabajadores que están en cada estación [28].

Han existido muchas investigaciones sobre la líneas de producción y las líneas de ensamblaje, sin embargo, existe poca investigación de ambas líneas en conjunto. Cuando se habla de las líneas de ensamblaje comúnmente se consideran la cantidad de piezas totales que ya fueron establecidas después de periodos largos de tiempo en la línea de producción.

El tema central de esta investigación es que la línea de ensamblaje considere lo que la línea de producción realiza en periodos de tiempo cortos, es decir, las piezas que surgen de la línea de producción que pueden ser utilizadas para comenzar a ensamblar los productos y el tiempo en el que surgen estas piezas de acuerdo a la organización especificada en la línea de producción, dependiente de los tiempos que requiere realizar los lotes de las piezas necesarias para los productos demandados [19].

Para establecer una conexión entre la línea de producción y la línea de ensamblaje es necesario considerar el tiempo que requiere la línea de producción para obtener una cantidad de cada tipo de piezas que sea necesario para ensamblar los productos demandados, es decir, no esperar a que se termine toda una jornada laboral sino considerar el periodo de tiempo que un molde requiere para lograr producir ciertas piezas de algún tipo, considerando los tiempos de preparación al montar un molde en una máquina [10, 12]. Es por eso que en esta investigación se considera el trabajo que lleva a cabo la línea de producción en conjunto con la línea de ensamblaje, de manera que exista un proceso que sea más parecido a lo que sucede en una compañía.

1.1 Hipótesis

La hipótesis que se considera para esta investigación es que el tiempo total que requiere una línea de ensamblaje para realizar la demanda de productos finales

disminuye cuando se toma en cuenta el trabajo realizado por la línea de producción a la par con la línea de ensamblaje.

1.2 Objetivo

Implementar algoritmos heurísticos que sean capaces de simular y optimizar la cantidad de productos finales que una línea de ensamblaje debe obtener al terminar una jornada laboral dependiendo de la producción de piezas que surgen de la línea de producción para así lograr obtener una minimización de tiempos de ocio en las estaciones de trabajo.

1.3 Estructura de la Tesis

En esta tesis se presenta la investigación que se llevó a cabo para lograr comprobar la hipótesis propuesta. En el capítulo 2 se presenta la descripción del Problema Producto-Pieza-Molde-Máquina [14] que tiene una organización factible con respecto a la producción de piezas necesarias para los productos demandados. Este problema entrega como respuesta la cantidad de piezas que surgen al terminar una jornada laboral con lo cual se puede seccionar esa jornada en periodos más cortos de tiempo y poder obtener la cantidad de piezas que surgen por periodo. Se incluye también en este capítulo una discusión de la literatura de los trabajos más relacionados a la presente tesis.

En el capítulo 3 se mencionan los intereses que existen en una línea de ensamblaje con respecto a la línea de producción para tener una buena organización de tiempos y que se mejore su desempeño en su asignación de ensambles. Se muestran también ejemplos sobre lo que la línea de ensamblaje realiza y el por qué es importante que ambas se lleven a cabo a la par. Se incluye también el planteamiento del Problema de Línea de Ensamblaje

En el capítulo 4 se muestran los algoritmos propuestos que simulan y optimizan del Problema de Línea de Ensamblaje, los cuales tienen como respuesta la organización de tiempos para ensamblar las piezas necesarias de los productos demandados dependiendo de la información obtenida sobre la cantidad de piezas realizadas en una línea de producción (información que se obtiene del Problema Producto-Pieza-Molde-Máquina). Se muestra además el algoritmo en el cual se procesa la información del Problema Producto-Pieza-Molde-Máquina, la cual es necesaria para llevar a cabo la simulación de la línea de ensamblaje.

En el capítulo 5 se muestran los resultados obtenidos a partir de la experimentación de los distintos algoritmos propuestos y se explica a detalle lo que se fue obteniendo de ellos con respecto a los intereses que se desean analizar para ver el funcionamiento de la fusión de tiempos de la línea de producción con la línea de ensamblaje.

Por último, en el capítulo 6 se muestran las conclusiones de los resultados de la investigación y el trabajo que se puede realizar en un futuro para darle seguimiento a la investigación propuesta.

Capítulo 2

Antecedentes y Revisión Bibliográfica

En este capítulo se muestra la revisión de antecedentes y literatura científica sobre artículos, investigaciones y problemas relacionados con un problema de una línea de producción en una empresa dedicada a la inyección de plásticos llamado el Problema de Producto-Pieza-Molde-Máquina a partir de la sección 2.1, el cual tiene como finalidad disminuir costos que generan los productos a realizar y busca obtener una buena asignación pieza-molde y molde-máquina para obtener esta minimización de costos; y con un problema de una línea de ensamble en la sección 2.3 en donde se busca, principalmente, minimizar el tiempo de ocio entre estaciones de trabajo teniendo una buena asignación de tareas en sus estaciones de trabajo.

2.1 Problema de Producto-Pieza-Molde-Máquina

El Problema de Producto-Pieza-Molde-Máquina (PPMM) se enfoca en la línea de producción de empresas que se dedican a realizar productos cuyas piezas necesarias se obtienen por medio de la inyección de plástico. Este problema determina el

tamaño de lote que surge de cada producto en un periodo y además la asignación pieza-molde-máquina adecuada en ese periodo para obtener ese lote [18]. Para poder realizarse una asignación factible pieza-molde-máquina se determina la compatibilidad que una pieza tiene hacia un molde y la compatibilidad que un molde tiene hacia una máquina, como se muestra en la figura 2.1 donde se puede observar el tipo de piezas que un producto necesita, la compatibilidad de las piezas hacia un molde y la compatibilidad del molde hacia una máquina. Además se consideran las capacidades de cada uno de los moldes de tal manera que se encuentre una solución óptima [21].

Este problema tiene como finalidad maximizar la ganancia que generan los lotes de los productos que se obtienen al final de un periodo considerando que existe un costo por mantener ciertas piezas en almacenamiento para que sean utilizadas en el siguiente periodo y un costo al retirar un molde de una máquina y colocar uno nuevo [3].

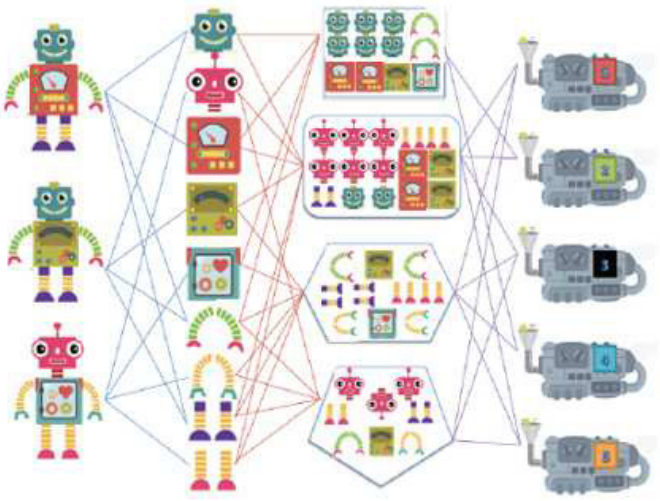


Figura 2.1: Asignación producto-pieza-molde-máquina [15].

2.2 Modelo Matemático del Problema Producto-Pieza-Molde-Máquina

Primeramente, se presenta el modelo matemático establecido por Ríos-Solís, et al. [27], el cual busca maximizar el beneficio que se obtiene al producir la mayor cantidad de lotes de productos en un periodo determinado, considerando piezas que se obtienen de una producción basada en la inyección de plástico. Lo que logra que se maximice este beneficio es la optimización de la asignación producto-pieza-molde-máquina, la cual tiene como característica principal la compatibilidad que tienen las piezas necesarias con los moldes disponibles y los moldes con las máquinas que están en funcionamiento, estudiado por Ibarra-Rojas, et al. [14].

Otros artículos relacionados con este modelo son el trabajo que presentan Chacón-Mondragón, et al. [8] que manejan una buena asignación pieza-molde-máquina en una planeación de producción en base al modelo matemático; y el trabajo que presentan Ibarra-Rojas, et al. [15] que en base al modelo matemático presentan una heurística para la organización de un problema de producción de inyección de moldes.

Este modelo representa una solución de una línea de producción, la cual será dato de entrada para los algoritmos propuestos de una línea de ensamblaje. Los algoritmos propuestos (capítulo 3 y 4) procesan la solución obtenida por este modelo y generan una programación para la línea ensamblaje.

Para este modelo se denota como I al conjunto de productos que son demandados. Se tiene el conjunto J, el cual denota los tipos de piezas distintos que son necesarias para realizar los productos. Se denota como K al conjunto de los moldes que son utilizados para producir las piezas necesarias y se tiene como $K(j)$ al subconjunto de moldes en donde el tipo de pieza j puede ser producida. Por último, se denota como L al conjunto de máquinas en las que los moldes son montados y a $L(k)$ al subconjunto de maquinas que es compatible con un molde k.

Tal como Ríos-Solís, et al. [27] mencionan en su investigación, se denota T como el conjunto de periodos considerado en la planeación. Se tiene también que el beneficio de un producto $i \in I$ por cada periodo $t \in T$ se considera como p_i^t y la demanda como d_i^t. El número de piezas de tipo j que se necesitan para armar una unidad de producto i se denota como a_{ij}. También, una pieza de tipo j puede ser producida utilizando un molde $k \in K(j)$ con un número de cavidades c_{jk}. Además, el tiempo que requiere realizar un montón de piezas tipo j con un molde $k \in K(j)$ instalado en una máquina $l \in L(k)$ se denota por b_{jkl}. El tiempo para montar y desmontar un molde k de una máquina $l \in L(k)$ se denota por s_{kl}. Se denota por h_j^t al costo de inventario que genera una pieza tipo j que no es utilizada durante un periodo t. Por último, se tiene también que para cada periodo $t \in T$ la máquina l tiene un tiempo de producción limitado f_l^t.

A continuación se definen las variables de decisión que se consideran para el modelo matemático del Problema Producto-Pieza-Molde-Máquina.

- x_{jkl}^t: número de lotes de la pieza j producidas con el molde k en la máquina l en el periodo t.
- w_i^t: tamaño de lote del producto i que es armado en el periodo t.
- r_j^t: número de piezas de tipo j que no fueron utilizadas al final del del periodo t. Estas piezas son puestas como inventario para el siguiente periodo, se considera $r_j^0 = 0$).
- y_{kl}^t: variable binaria que toma valor 1 si el molde k es utilizado en la máquina l en el periodo t, 0 en otro caso.
- z_{kl}^t: variable binaria que toma valor 1 si el molde k es montado en la máquina l entre el periodo t y el periodo $t+1$, 0 en otro caso.

Como modelo matemático del Problema de Producto-Pieza-Molde-Máquina se tiene la siguiente función objetivo [27].

$$\text{máx} \sum_{t \in T} \left(\sum_{i \in I} p_i^t w_i^t - \sum_{j \in J} h_j^t r_j^t \right) \tag{2.1}$$

Sujeta a:

$$w_i^t \leq d_i^t \qquad t \in T, i \in I \tag{2.2}$$

$$\sum_{j \in J(k)} x_{jkl}^t \leq M_{kl}^t y_{kl}^t \qquad t \in T, k \in K, l \in L(k) \tag{2.3}$$

$$\sum_{l \in L(k)} z_{kl}^t \leq 1 \qquad t \in T, k \in K \tag{2.4}$$

$$r_j^t + \sum_{i \in I(j)} a_{ij} w_i^t = r_j^{t-1} + \sum_{k \in K(j)} \sum_{l \in L(k)} c_{jk} x_{jkl}^t \qquad t \in T, j \in J \tag{2.5}$$

$$\sum_{k \in K(l)} \left(s_{kl}(y_{kl}^t - z_{kl}^{t-1}) + \sum_{j \in J(k)} b_{jkl} x_{jkl}^t \right) \leq f_l^t \qquad t \in T, l \in L \tag{2.6}$$

$$r_j^t = z_{kl}^t = 0 \qquad t = |T|, j \in J, k \in K, l \in L(k) \tag{2.7}$$

$$w_i^t, x_{jkl}^t, r_j^t \in \mathbb{Z}^+ \qquad t \in T, i \in I, j \in J, k \in K(j), l \in L(k) \tag{2.8}$$

$$z_{kl}^t, y_{kl}^t \in \{0, 1\} \qquad t \in T, k \in K, l \in L(k) \tag{2.9}$$

La función objetivo (2.1) de este modelo matemático busca maximizar el beneficio que se obtiene de cada lote de producto terminado en un periodo considerando que se tiene un costo de almacenamiento de piezas sobrantes que se utilizarán en el siguiente periodo. Las restricciones (2.2) consideran que el tamaño de lote de cada producto no puede sobrepasar la demanda que se tiene del mismo. En cuanto a las restricciones (2.3) se tiene que solamente pueden haber lotes de piezas tipo j hechas en el molde k en la máquina l si el molde k está realmente montado en la máquina l, es decir, si el molde k está activo en la máquina l entonces podrá realizar lotes de piezas tipo j. Las restricciones (2.4) nos dicen que el molde k solamente puede ser montado en una máquina l en el periodo t. Las restricciones (2.5) mantienen un equilibrio entre el inventario del periodo $t-1$ más el total de piezas que se producen en el periodo t y las piezas de inventario en el periodo t más el total de piezas utilizadas para obtener los productos. Por último, las restricciones (2.6) tienen como fin que el tiempo que toma instalar un molde k en una máquina l más el tiempo que toma producir un lote no debe exceder el tiempo total que tiene permitido la máquina l durante un periodo t.

Para lograr unir la línea de producción con la línea de ensamblaje se busca hacer un análisis de la información que la línea de producción necesita dado que se tiene un tiempo de ciclo flexible en la línea de ensamblaje y se requiere conocer los tiempos y asignaciones que la línea de producción realiza, como lo mencionan con más detalle Lapierre, Ruiz y Soriano [17]. Este modelo es propuesto dado que la solución que se genera a partir de él es un dato de entrada para los algoritmos que se proponen en los siguientes capítulos, los cuales estudian el Problema de Línea de Ensamblaje. La solución dada por este modelo es el tiempo de ciclo en el cual una cantidad de algún tipo de piezas se realiza en la línea de producción, además de la asignación pieza-molde-máquina que existe en la línea de producción y el tiempo total que un molde está colocado en una máquina.

2.3 Línea de ensamblaje

En una línea de ensamblaje, de acuerdo con Jaramillo-Garzon y Restrepo-Correa [16], se tiene como propósito distribuir las tareas a realizar a las distintas estaciones de trabajo en base a un objetivo, ya sea maximizar eficiencia de la línea o minimizar tiempos de ocio entre estaciones, con las restricciones de que cada tarea debe de asignarse solo a una estación de trabajo y que la suma de los tiempos de las tareas asignadas a cada estación no deben exceder al tiempo de ciclo establecido para cada estación.

Para poder realizar una buena asignación de tareas entre estaciones de trabajo con el objetivo de que se reduzcan los tiempos de espera entre el ensamble de un producto y otro se requiere realizar un balance en la línea de ensamblaje, lo cual involucra el análisis de la cantidad de estaciones de trabajo, el número de operadores que deben involucrarse en cada estación y, en la mayoría de los casos, el flujo que debe existir entre una estación y otra (lo cual puede ser flujo de piezas o flujo de productos) [4].

Estos problemas de balanceo de la línea de ensamblaje se dividen en dos categorías, el primero son los Problemas de Balanceo de una Línea de Ensamblaje Simple, (SALBP por sus siglas en inglés), los cuales son problemas simples en donde se consideran pocas variables de entrada para reducir la complejidad del balanceo. Por otro lado, se tienen a los Problemas de Balanceo de una Línea de Ensamblaje General (GALBP por sus siglas en inglés), los cuales son problemas generales de balanceo de la línea que estudian casos más reales y complejos que se presentan en el área industrial, como lo mencionan Jaramillo-Garzon y Restrepo-Correa [16].

Para el interés que se tiene en esta investigación se llevará a cabo un problema de balanceo simple de una línea de ensamblaje y para problemas de balanceo se considera realizar heurísticas que solucionen el balanceo de la línea debido a la cantidad de variables y actividades involucradas, como lo mencionan Medina-Chacón

e Illada-García [7].

Existen distintos procedimientos heurísticos que resuelven el problema del balanceo simple de una línea de ensamblaje. Una de ellas son las heurísticas de una sola pasada, las cuales usan reglas de decisión simples que resuelven problemas con disposición lineal de estaciones de trabajo y la secuencia de la fabricación, como lo mencionan Pinzón-Salazar y Santa-Luna [26], lo cual es de interés para la presente investigación.

También se tienen las heurísticas tipo voraz, las cuales se caracterizan por utilizar los datos del problema construyendo paso a paso una solución al tomar una decisión; este tipo de heurísticas propone un procedimiento bajo una búsqueda dispersa como lo mencionan Bautista, Fernández y González [1]. Este tipo de heurísticas también se utilizan para resolver problemas multi-objetivo que se basan en la optimización de las colonias de hormigas; también se tienen algoritmos voraces de búsqueda aleatoria para resolver problemas más realistas como lo mencionan Chica, Cordón y Damas [9].

Para llevar a cabo la solución de estos problemas existen dos supuestos muy importantes que se consideran en la línea de ensamblaje. El primero de ellos es que hay un tipo de ritmo establecido, es decir, si una estación de trabajo comienza cada cierto periodo de tiempo ensamblando un tipo de pieza j (y tarda justo ese periodo en ensamblarla) entonces todas las estaciones de trabajo tomarán esa misma cantidad de periodo de tiempo en ensamblar ese tipo de pieza j, a esto se le llama *tiempo de ciclo*, el cual, de acuerdo a la literatura, puede ser una unidad de tiempo en el algoritmo propuesto, como lo mencionan Yano y Bolat [32].

El segundo supuesto es que el procesamiento permitido es equivalente para todas las estaciones de trabajo, es decir, todas las estaciones de trabajo de la línea de ensamblaje pueden ensamblar cualquier producto i que cuente con piezas disponibles del tipo que necesita, para lo cual se requiere de múltiples estaciones si se desea un balanceo eficiente en la línea de ensamblaje como lo menciona Buxey [6].

Murillo-García, et al. [20], mencionan que pueden existir más de un trabajador en cada estación de trabajo que realicen la misma tarea, lo cual ayuda a que estos elementos representen pequeñas porciones del trabajo total que se debe desempeñar en cada estación de trabajo. Así, el tiempo que un trabajador calificado tarda en realizar una tarea utilizando un método establecido, a una velocidad promedio y en condiciones normales es el tiempo de ciclo que se establece para llevar a cabo las tareas en las diferentes estaciones de trabajo, como lo mencionan Peña-Orozco, Neira-García y Ruiz-Grisales [24].

Uno de los objetivos secundarios de esta investigación es mostrar que se puede aumentar la cantidad de productos finales que surgen en la línea de ensamblaje y para ello se espera que no existan amontonamientos de las piezas que surgen de la línea de producción en las estaciones de trabajo de la línea de ensamblaje, para que haya una continuidad entre un ensamble y otro de los productos, así tendremos que no deben existir estaciones de trabajo en donde se tengan muchas piezas en espera, como se muestra en la figura 2.2 [11]. Otra de las razones para que se ensamble la mayor cantidad de productos es que no existan piezas sobrantes que queden en almacenamiento ya que esto genera costos que se desean evitar.

Figura 2.2: Amontonamiento de piezas en una línea de ensamblaje.

En el siguiente capítulo se describe el Problema de Línea de Ensamblaje, el cual es el problema que se estudia en esta tesis. Se muestra un algoritmo que sirve como base para entender mejor el problema y con el cual se resolvió un ejemplo

que es apoyo para lograr tener de manera visual las variables que se requieren como entrada y la respuesta de salida de nuestro algoritmo propuesto.

Capítulo 3

Problema de Línea de Ensamblaje

En este capítulo se presenta a detalle el Problema de Línea de Ensamblaje que se estudia en esta tesis. Primeramente, se describe el problema que se desea resolver, de manera general se plantea cómo funciona el algoritmo propuesto y las decisiones que se toman para obtener una solución al problema y la complejidad que éste tiene. En segunda instancia se presenta un primer ejemplo junto con un algoritmo base para detallar el Problema de Línea de Ensamblaje y se muestra un ejemplo particular que se resuelve con el algoritmo base propuesto. En el capítulo 4 se presentan los algoritmos para resolver este problema de manera más apegada a la realidad.

Una vez que el modelo del Problema de Producto-Pieza-Molde-Máquina se resuelve, es decir, una vez que se obtiene el plan de producción de las piezas y se obtiene la respuesta sobre el periodo en el cual se producen, el tipo de pieza que surge, y el tamaño de lote que se obtiene, se guarda esa información para que sea utilizada después en el Problema de Línea de Ensamblaje. Además, se obtiene la asignación pieza-molde-máquina que se utiliza para la producción de estos lotes y la información sobre los tiempos que tardan en producirse los montones de piezas. A continuación se tiene de manera gráfica una respuesta al Problema Producto-Pieza-

Molde-Máquina.

Considerando el caso en el que se tienen tres máquinas en funcionamiento, cuatro moldes distintos y once tipos de piezas que pueden ser producidas en un periodo $t \in T$, se muestra en la figura 3.1 una respuesta sobre la asignación pieza-molde-máquina que se requiere para obtener un lote de cada producto $i \in I$.

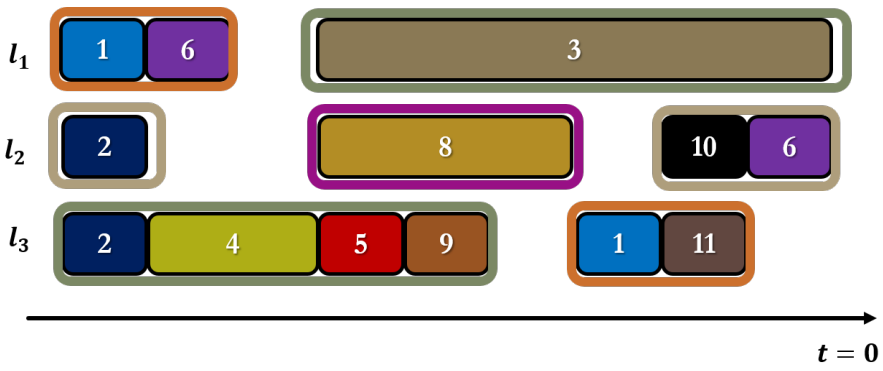


Figura 3.1: Respuesta gráfica del Problema Producto-Pieza-Molde-Máquina.

Se muestra cada una de las máquinas l_1, l_2, l_3 que están en funcionamiento durante el periodo $t \in T$. Cada uno de los bloques que tiene asignado un número es un lote de una pieza tipo $j \in J$ donde j toma valores desde uno hasta once, y los colores que rodean a estos bloques de lotes son los distintos moldes que fueron montados en las máquinas. Esta respuesta del Problema Producto-Pieza-Molde-Máquina (mostrada en la figura 3.1) es para un periodo $t = 0$.

El propósito del Problema de Línea de Ensamblaje es tomar en cuenta la respuesta del Problema Producto-Pieza-Molde-Máquina, tanto el número de lotes de piezas que surgen por periodo como la asignación de pieza-molde-máquina, a fin de que se pueda realizar una asignación de tiempos en la organización de la línea de ensamblaje.

3.1 Descripción del Problema de Línea de Ensamblaje

El problema de Línea de Ensamblaje tiene como objetivo minimizar los tiempos de ocio en las estaciones de trabajo. Para poder lograr esta minimización se pretende tener una organización factible de tiempos para ensamblar los productos demandados, lo cual se tiene al momento de que todos los tipos de productos se ensamblan total o parcialmente.

Para nuestro Problema de Línea de Ensamblaje se denota como t a los periodos de tiempo $t \in T$ considerados en la línea de producción; se denota como p a los periodos de tiempo $p \in P$ considerados en el Problema de Línea de Ensamblaje, los cuales son intervalos de tiempo menores a los periodos de tiempo t; y se denota también como x_{jkl}^{tp} a la cantidad de piezas de cada tipo j totales que se obtienen de la línea de producción en un periodo p de un molde k montado en una máquina l en un periodo t de la línea de producción, la cual es nuestra variable de decisión.

La función objetivo de nuestro Problema de Línea de Ensamblaje es maximizar la cantidad de productos finales de cada tipo i al terminar un periodo t en el Problema Producto-Pieza-Molde-Máquina, considerando una penalización por piezas que quedan en el almacenamiento al finalizar el periodo t, ya que al maximizar la cantidad de productos finales se tendrá en consecuencia una minimización en los tiempos de ocio de la línea de ensamblaje.

Para que comience una organización en este problema se tiene un procesamiento de la información del Problema Producto-Pieza-Molde-Máquina, este procesamiento de información considera la información de las instancias que son utilizadas para el Problema Producto-Pieza-Molde-Máquina y la respuesta que surge de dicho problema.

Así, se toma la información de la cantidad total de productos i, la cantidad de

piezas tipo j que necesita cada producto i, el tipo de pieza j que cada producto i necesita, la demanda de cada producto i y la preferencia de ensamble que se tiene de cada producto i. Esta información se extrae de las instancias del Problema Producto-Pieza-Molde-Máquina y se utiliza como información de entrada para el Problema de Línea de Ensamblaje.

También, al realizarse un procesamiento de la respuesta del Problema Producto-Pieza-Molde-Máquina se obtiene como entrada para el Problema de Línea de Ensamblaje la cantidad de periodos p que se tienen en total dependientes de la cantidad de periodos t en que las máquinas estuvieron en funcionamiento en la línea de producción, la cantidad de ensambles que se pueden realizar por periodo p de cada pieza tipo j ($cantEnsambles_{ij}^{p}$) y la cantidad de piezas tipo j que surgen por periodo p de la línea de producción ($piezasPeriodo_{j}^{p}$).

Por cada periodo de tiempo p del Problema de Línea de ensamblaje se verifica la cantidad de piezas tipo j que surgen de la línea de producción, por ejemplo, si se considera una respuesta como la que se muestra en la figura 3.2 para el periodo $p = 1 = 1$ se tiene que al finalizar el periodo $p = 1$ el Problema de Línea de Ensamblaje tendrá a su disposición la cantidad de piezas $piezasPeriodo_{j}^{p=1} = x_{jk0}^{t1} + x_{jk1}^{t1} + x_{jk2}^{t1} + ... + x_{jkn}^{t1}$, con las cuales se decide si se puede comenzar a ensamblar algún producto i dependiendo de si está disponible la cantidad necesaria de piezas tipo j. Así, para el periodo $p = 2$ se obtiene que la Línea de Ensamblaje tendrá a su disposición la cantidad de piezas $piezasPeriodo_{j}^{2} = x_{jk0}^{t2} + x_{jk1}^{t2} + x_{jk2}^{t2} + ... + x_{jkn}^{t2}$ y se decide a partir de la suma de x_{jkl}^{t2} con las piezas sobrantes en el periodo $p = 1$ qué productos i pueden comenzar a ensamblarse; y así sucesivamente hasta que se llega al periodo $p = n$ en donde se considera a disposición de la línea de ensamblaje la cantidad de piezas $piezasPeriodo_{j}^{n}$ adicional a las piezas que sobraron en los periodos $p = 1, p = 2, ...p = n - 1$ y se decide a partir de esa cantidad los productos i que se van a ensamblar en el periodo $p = n$.

Una vez que se obtiene la cantidad de piezas disponibles de la línea de produc-

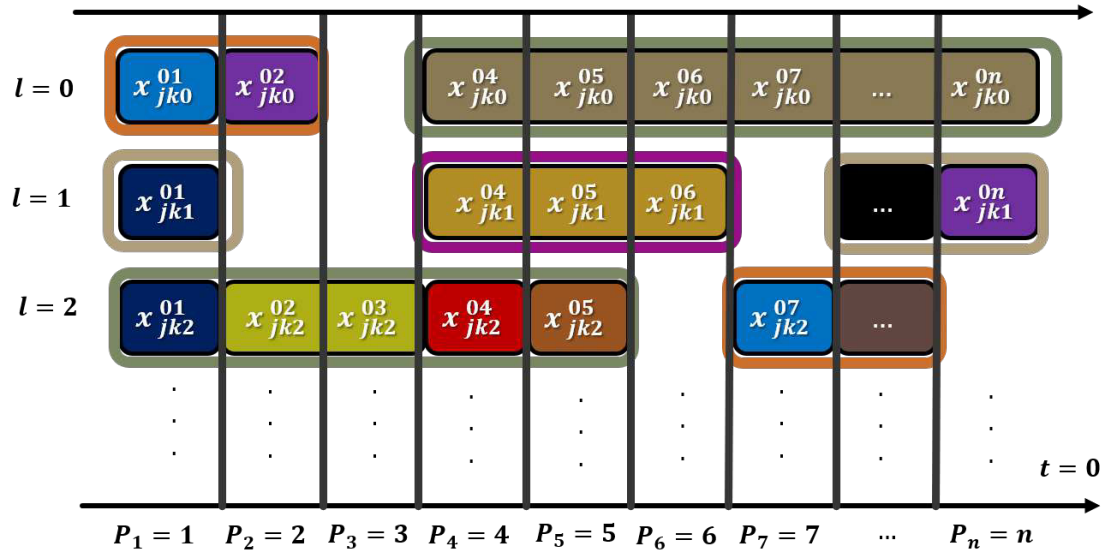


Figura 3.2: Respuesta gráfica del Problema Producto-Pieza-Molde-Máquina seccionado en periodos p considerados para el Problema de Línea de Ensamblaje.

ción en un periodo p ($piezasPeriodo_j^p$) se va verificando por orden de prioridad de los productos i cuáles son los que se pueden comenzar a ensamblar en el periodo p ya sea total o parcialmente.

Al seleccionarse un producto i, con el propósito de comenzar a ser ensamblado, se calcula la cantidad de piezas que podrían ser ensambladas en el periodo p ($prodEnsamble_{ij}$) dependiente a la cantidad de piezas tipo j que necesita un producto i y la cantidad de piezas de ese tipo que surgieron de la línea de producción en el periodo p; si la cantidad de piezas que pueden ser ensambladas en una estación de trabajo no supera a la cantidad de piezas que pueden ser ensambladas de acuerdo a las piezas disponibles ($prodEnsamble_{ij} \geq cantEnsamble_{ij}^p$) entonces se ensambla la cantidad de piezas $cantEnsamble_{ij}^p$ y esta cantidad disminuye en $piezasPeriodo_j^p$.

Cabe señalar que no siempre se tiene que se pueden ensamblar todas las piezas obtenidas en un periodo p dado que se tienen restricciones sobre el tiempo en que tarda una estación de trabajo en ensamblar una cantidad de piezas tipo j de algún producto i y sobre la cantidad de estaciones de trabajo que están en funcionamiento

en la línea de ensamblaje. Para ello, se tiene una penalización en cuanto a las piezas totales que quedan en el almacenaje al finalizar un periodo t.

Para que se pueda llevar a cabo esta organización de tiempos se tiene una optimización de un problema de optimización combinatoria, el cual se simula y optimiza mediante una heurística propuesta en esta investigación. Esta heurística propuesta es de tipo constructivo voraz semi-aleatorio dado que existe una lista restringida de candidatos que selecciona a los mejores candidatos a ingresar en la solución y de la cual aleatoriamente se elige a uno de ellos. Cada vez que las piezas de un producto i se terminan de ensamblar se va ingresando el producto en la solución.

Las decisiones que se toman en el Problema de Línea de Ensamblaje es qué productos son los que se van a ensamblar en un periodo p y qué cantidad de piezas se ensamblan, y en base a ello se va reduciendo el tiempo de ocio que las estaciones de trabajo podrían tener entre el ensamblado de un producto y otro, lo cual es el objetivo de este problema.

El tema de la complejidad computacional del Problema de Línea de Ensamblaje es interesante pero no se estudia a detalle en esta tesis. Nos parece que el Problema de Línea de Ensamblaje es no polinomial duro (NP-Difícil) de acuerdo a las distintas variables que se utilizan para resolverlo (que cuentan con un rango n entre sus posibles valores). En efecto, el Problema de Línea de Ensamblaje está relacionado con el problema de asignación generalizada tal como lo mencionan Ocampo, Toro y Echeverry [23]. En el marco de este estudio, una heurística es una buena propuesta para la reducción del tiempo computacional que esta optimización requiere.

De acuerdo con la búsqueda bibliográfica realizada la diferencia que existe entre las investigaciones que otros autores han realizado de la línea de ensamblaje y el Problema de Línea de Ensamblaje, estudiado en esta tesis, es que otros autores consideran los lotes totales que surgen de la línea de producción y en base a esa cantidad se van repartiendo las tareas a las estaciones de trabajo como lo mencionan Restrepo, Medina y Cruz [25], mientras que el Problema de Línea de Ensamblaje

tiene una conexión entre la línea de producción del Problema Producto-Pieza-Molde-Máquina y la línea de ensamblaje del Problema de Línea de Ensamblaje donde realizan sus procesos al mismo tiempo.

3.2 Algoritmo Base

En un principio, para comenzar la investigación del Problema de Línea de ensamblaje, se tiene que cada uno de los productos necesita la misma cantidad de tipo de piezas necesarias (es decir, que cada tipo de productos necesita de n tipos diferentes de piezas para su ensamblaje) para lo cual se requiere de vectores que contengan la información sobre el tipo de pieza de cada uno de los productos ($tipoPiezas$), la cantidad de piezas que necesitan los productos de cada tipo de pieza ($cantPiezas$), la cantidad de piezas que se pueden ensamblar de cada tipo de pieza ($cantEnsambles$) y la cantidad de piezas que van saliendo de la línea de producción por cada periodo p ($piezasPeriodo$), como se muestra en el algoritmo 1 propuesto en esta tesis. Este algoritmo representa una heurística del tipo voraz constructivo [31] que se enfoca en ir construyendo una solución factible que se va generando al elegir el mejor candidato cada vez que se toma una decisión y éste se agrega a la solución. Los distintos tipos de productos que son demandados se registran en el vector $productos$.

Cada vez que pasa un periodo p en la línea de producción se verifica para cada producto i si existen piezas suficientes de todos los tipos j de piezas que ese producto i necesita para poder ensamblar toda la demanda que se requiere (donde i se toma de manera creciente) como se muestra en la línea ocho de nuestro algoritmo. Una vez que se tienen las piezas suficientes de ese producto i entonces se ensamblan, se verifica por cada tipo de piezas j si la cantidad de piezas a ensamblarse es mayor que la cantidad que se pueden ensamblar en ese periodo p, como se muestra en la línea diez. Si la cantidad de piezas que se requieren ensamblar para cubrir la demanda es mayor que la cantidad de piezas que se pueden ensamblar entonces se ensambla solo la cantidad de piezas que se puedan y el periodo p aumenta en uno, como se muestra

en las líneas ocho y nueve del algoritmo 1; si no, entonces se considera que la línea de ensamblaje solamente tarda un periodo en ensamblar esa pieza j y se ensambla solamente la cantidad de piezas que se necesitan ensamblar como se muestra de la línea diez a la quince.

Una vez que se tiene que todos los productos demandados son ensamblados entonces se aumenta en uno la cantidad de productos terminados lo cual se guarda en una variable llamada $prodTerminados$, una vez terminado un producto i se verifica si en los periodos p que pasaron mientras se ensamblaba ese producto se produjeron algunas piezas en la línea de producción y se hace un conteo junto con las piezas registradas anteriormente, esto se aumenta en la variable $piezasPeriodo$, como se muestra en la línea diecisiete. Este proceso se repite hasta un determinado número de periodos p registrado en la variable $periodosTotales$ y la respuesta que se da es cuáles productos lograron realizar su demanda total.

Este algoritmo es una base para las heurísticas que se presentan más adelante en este trabajo, las cuales son utilizadas para realizar los experimentos necesarios para verificar si es mejor un ensamblaje total a un ensamblaje parcial en cuanto a la maximización de la cantidad de productos finales. Además, es un algoritmo que es apoyo para obtener respuestas básicas y entender cómo funciona la heurística constructiva para un ejemplo en particular del Problema de Línea de Ensamblaje.

3.3 Ejemplo del Problema de Línea de Ensamblaje

Consideremos el caso que se muestra en la figura 3.1 como respuesta de la línea de producción para un ejemplo en el cual se tienen cuatro productos distintos i que son demandados (de esta manera tenemos que $i \in \{1, 2, 3, 4\}$), de los cuales se tiene una demanda de 12 para el producto tipo 1, de 17 para el producto tipo 2, de 14 para el producto tipo 3 y de 14 para el producto tipo 4. Además, se toma en cuenta

Algoritmo 1 Ensambles de productos con igual cantidad de piezas necesarias para su ensamblaje

Input: $productos, cantPiezas, tipoPiezas, cantEnsambles, piezasPeriodo$

Output: productos ensamblados ($prodTerminados$)

1: $p \leftarrow 0$
2: **repeat**
3: $p \leftarrow p + 1$
4: **for** $i \in \{1, ..., productos\}$ **do**
5: **if** $piezasPeriodo_i^p[\alpha] \geq cantPiezas_i[\alpha], \ \forall \alpha \in tipoPiezas_i$ **then**
6: **for** $\alpha \in \{1, ..., \mid tipoPiezas_i \mid\}$ **do**
7: **if** $cantPiezas_i[\alpha] \geq cantEnsambles_i[\alpha]$ **then**
8: **while** $cantPiezas_i[\alpha] > 0$ **do**
9: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $cantEnsambles_i[\alpha]$
10: $p \leftarrow p + 1$
11: **end while**
12: **else**
13: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $cantEnsambles_i[\alpha]$
14: **end for**
15: $prodTerminados \leftarrow prodTerminados \cup i$
16: **end for**
17: $piezasPeriodo \leftarrow piezasPeriodo \cup piezasPeriodo^p$
18: **until** $p < periodosTotales$
19: ***return*** $prodTerminados$

que en un mismo periodo p pueden ensamblarse los distintos tipos de pieza de los productos. También como supuesto tenemos que las piezas 9, 10 y 11 son necesarias para comenzar el ensamblaje de los productos que las necesitan ya que son las piezas base para los distintos productos.

Para cada producto i se tienen distintos tipos de piezas que necesita, en este caso se considera que cada uno de esos productos se compone de cuatro tipos de piezas diferentes de cada tipo de producto como se muestra en la primer columna de la tabla 3.1. Así para el producto tipo 1 tenemos que necesita del tipo de pieza 2, 5, 8 y 10; para el producto tipo 2 se considera que necesita de las piezas tipo 1, 3, 6 y 11; con respecto al producto tipo 3 se necesitan las piezas tipo 2, 3, 5 y 9; por último, se tiene que el producto tipo 4 necesita del tipo de pieza 1, 4, 6 y 9.

Ahora, una vez teniendo qué tipo de pieza necesita el producto se debe tomar en cuenta también qué cantidad de piezas necesita cada uno de ellos dependiendo de la demanda que se requiere. Se muestra también en la tabla 3.1, en la segunda columna de cada tipo de producto, el total de piezas necesarias de cada tipo conforme a la demanda de los productos, de esta manera tendremos que el producto tipo 1 necesita 24 piezas de tipo 2, 12 piezas de tipo 8, 24 piezas de tipo 5 y 12 piezas de tipo 10; el producto tipo 2 necesita 34 piezas de tipo 1, 17 piezas de tipo 3, 34 piezas de tipo 6 y 17 piezas de tipo 11; de la misma manera para el producto 3 y 4.

Producto 1		Producto 2		Producto 3		Producto 4	
Tipo	Cantidad	Tipo	Cantidad	Tipo	Cantidad	Tipo	Cantidad
2	24	1	34	2	28	1	28
8	12	3	17	3	14	4	14
5	24	6	34	5	28	6	28
10	12	11	17	9	14	9	14

Tabla 3.1: Tipos de pieza y cantidad de piezas necesarias para ensamblar los productos.

Obteniendo la información de la respuesta gráfica del Problema Producto-

Pieza-Molde-Máquina mostrada en la figura 3.1 se tiene que, por ejemplo, en el periodo de tiempo P_1 se realizaron las piezas tipo 1 y tipo 2, en el periodo P_2 se realizaron las piezas tipo 6 y tipo 4, en el periodo P_3 se realizó la pieza tipo 4 y así sucesivamente hasta llegar a que en el periodo P_9 se realizaron las piezas tipo 3 y tipo 6. En la tabla 3.2 se muestra qué cantidad de pieza de cada tipo es la que surge en algunos periodos p, una vez establecida esta información se puede proceder a buscar una solución para el problema. Existen dos casos que se tomarán en cuenta para poder resolver el Problema de Línea de Ensamblaje.

Periodo p	Tipo de Pieza j	Cantidad de piezas j
1	1	36
1	2	60
2	6	36
2	4	8
3	4	8
⋮	⋮	⋮
9	3	6
9	6	36

Tabla 3.2: Respuesta del Problema Producto-Pieza-Molde-Máquina donde se muestra el tipo de pieza j y la cantidad de ese tipo de pieza que surge de la línea de producción en un periodo p.

El primer caso es que un producto i puede comenzar a ensamblarse solamente si existen suficientes piezas para cada tipo j que el producto i necesita, por ejemplo, si en el periodo P_6 se tiene que ya fueron producidas las piezas tipo 1, 4, 6 y 9 y que de cada una de ellas existe una cantidad de 36, 16, 36 y 32, respectivamente, entonces se tiene la posibilidad de que el producto tipo 4 se ensamble ya que se tienen las piezas suficientes de cada tipo para poder terminar la demanda de ese producto. Así una posible solución para este ejemplo sería que el producto tipo 4 comienza a ensamblarse en el periodo P_6, el producto tipo 3 comienza a ensamblarse en el

periodo P_7, el producto tipo 1 en el periodo P_9 y el producto tipo 2 en el periodo P_{10}, como se muestra en la tabla 3.3. Este primer caso se resuelve con el algoritmo 1, sin embargo es de interés también que los productos puedan ser sub-ensamblados y es por eso que se muestra el segundo caso, en donde se tienen sub-ensambles de los distintos productos i que son demandados en este ejemplo.

Periodo p	Producto i	Tipo de pieza j
6	4	1
6	4	4
6	4	6
7	3	2
7	3	3
7	3	5
9	1	2
9	1	8
9	1	5
10	2	1
10	2	3
10	2	6

Tabla 3.3: Respuesta de la línea de ensamblaje para ensambles totales.

Como segundo caso se tiene que pueden existir *sub-ensambles*, esto permite que si existen piezas suficientes para un tipo de pieza j (sin importar qué producto i lo requiera) entonces estas piezas comienzan a ensamblarse [22]. De esta manera tendremos que, por ejemplo, si para el periodo P_6 se produjeron las piezas tipo 9 entonces se puede verificar si los productos tipo 3 y 4 (los cuales tienen como base a la pieza tipo 9) tienen ya piezas suficientes para comenzar a hacer sub-ensambles con algún tipo de pieza que necesiten. Así, considerando a los productos tipo 3 y 4, tendremos que hay piezas suficientes en el periodo P_6 para ensamblar las piezas tipo 2 y 5 del producto tipo 4, y las piezas tipo 1, 4 y 6 para el producto tipo 4, de

la misma manera se verifica para los siguientes periodos y se obtiene como respuesta que en el periodo P_9 se hacen sub-ensambles del producto tipo 1 y 2 y en el periodo P_{10} se hacen sub-ensambles del producto tipo 2 y 3, como se muestra en la tabla 3.4.

Periodo p	Producto i	Tipo de pieza j
6	3	2
6	3	5
6	4	1
6	4	4
6	4	6
9	1	2
9	1	5
9	1	8
9	2	1
10	2	3
10	2	6
10	3	3

Tabla 3.4: Respuesta de la línea de ensamblaje para ensambles totales y sub-ensambles de productos.

Ambos casos tienen un resultado similar en cuanto al periodo p en donde se ensambla la demanda de los últimos producto, esto es por el supuesto que se tiene tan fuerte en un inicio en el Problema de Línea de Ensamblaje de poder ensamblar al mismo tiempo cualquier tipo de pieza de cualquier producto, es decir, que existen suficientes estaciones de ensamblado para cubrir cualquier ensamblaje.

En siguientes secciones se verán además casos en los cuales tenemos distintos productos y cada uno de ellos con distinta cantidad de tipo de piezas que necesitan.

Capítulo 4

Algoritmos para el Problema de Línea de Ensamblaje

Teniendo en cuenta qué es lo que se quiere hacer en la línea de ensamblaje y qué información se necesita de la línea de producción se llevó a cabo la programación del Problema de Línea de Ensamblaje. Para ello se construyeron los algoritmos que se presentan a continuación, los cuales se programaron en lenguaje `Python` y generamos nuestras propias instancias a partir de las instancias y resultados de la línea de producción, esto es, a partir del Problema de Producto-Pieza-Molde-Máquina, las cuales se obtuvieron realizando una programación en `R`.

4.1 Algoritmos para el Problema de Línea de Ensamblaje

Una vez que se estudió el hecho de pasar de la línea de producción a la línea de ensamblaje y que se tenía la idea de seccionar los periodos t de la línea de producción en periodos p en la línea de ensamblaje se realizaron algoritmos para resolver esta problemática, los cuales son una evolución del trabajo que se estuvo realizando y que se mostrarán a continuación. Estos algoritmos consideran la información la

información necesaria sobre los productos (es decir, el tipo de piezas que requieren, la cantidad de piezas que requieren de cada tipo de piezas y su demanda), la información sobre los tiempos en que los moldes son montados en las máquinas, los tiempos que toma la línea de producción en terminar de hacer un montón de piezas de algún tipo j en algún molde k montado en una máquina l, y la respuesta del Problema Producto-Pieza-Molde-Máquina como una instancia.

4.1.1 Algoritmo de Ensamblajes Totales

Ahora bien, se sabe que los productos que se elaboran en la industria no son iguales unos a otros, así que se debe considerar que un producto no necesariamente tiene que tener la misma cantidad de tipos de piezas que otro. Además, no es conveniente que un producto tenga que esperar a que todas las piezas que se requieren para cubrir su demanda se hayan realizado en la línea de producción, por lo que, en el algoritmo 2, de tipo voraz constructivo [31], tenemos que cada vez que pasa un periodo p, se verifica qué cantidad de demanda puede hacerse dependiendo de la cantidad de piezas que necesita un producto de cada tipo j. Si existen piezas suficientes para cubrir aunque sea un producto tipo i en un periodo p entonces éste se ensambla y se deja para después el resto de la demanda. Además, se considera que pueden realizarse ensambles parciales de los productos, es decir, sub-ensambles.

Para este algoritmo tenemos vectores en donde se verifica la demanda por producto, el tipo de piezas que se requieren para cada producto, la cantidad de productos que se requieren de cada tipo, la cantidad de piezas que se pueden ensamblar por periodo p, las piezas que se realizan en la línea de producción por periodo p y las piezas que hasta el momento se han ensamblado de cada producto.

Cada que pasa un periodo p se verifica para cada producto i (de manera ascendente) si existen piezas suficientes para elaborar al menos un producto de los demandados, por ejemplo, si existiera una demanda de ochocientos productos pa-

ra el tipo de producto uno pero se tienen piezas suficientes para realizar al menos uno de esos ochocientos productos entonces se puede comenzar la verificación de piezas para que se lleve a cabo el ensamble, como muestra la línea ocho. Una vez que se tiene que al menos existen piezas suficientes para realizar un producto de la demanda, se calcula para cada pieza tipo j del producto i el mayor entero menor que la división entre la cantidad de piezas tipo j que se pueden ensamblar en ese periodo p y la cantidad de piezas que se requieren de ese tipo j para el producto i, de esta manera obtenemos cuántos productos pueden ser ensamblados en ese periodo, registrado como la variable *prodEnsamble*, como se muestra en la línea diez.

Una vez que conocemos esta cantidad de demanda se verifica si la demanda del producto i sustrayéndole la cantidad de productos ensamblados con respecto a ese tipo de pieza j es mayor o igual a la cantidad de productos que se pueden ensamblar en ese periodo (lo cual se calcula en *prodEnsamble*) como se muestra en la línea once. Esto quiere decir, por ejemplo, que si tenemos una demanda de cien piezas tipo j y ya se han ensamblado ochenta, entonces quedan veinte productos con el tipo de pieza j por ensamblar, supongamos que estas piezas tipo j son las piezas tipo dos que se mencionaron en el ejemplo anterior, como se pueden ensamblar cinco productos tipo uno considerando el tipo de pieza dos entonces tendremos que la resta, el cual es veinte, es mayor que el valor de *prodEnsamble*, el cual es cinco, entonces verifica si la cantidad de piezas tipo j es suficiente para poder ensamblar esa cantidad de demanda. Si sí se puede entonces lo hace; si esto se cumple entonces se verifica que hayan piezas suficientes de la línea de producción que nos ayuden a ensamblar la demanda de *prodEnsamble*. Si existen piezas suficientes entonces ensambla esa cantidad de piezas y registra el total de demanda realizada en la variable *piezasTerminadas*, como se muestra de la línea doce a la catorce.

En cambio, si el resultado de sustraer la cantidad de productos ensamblados de la demanda requerida no es mayor o igual que la cantidad de demanda que puede realizarse en ese periodo entonces se tiene una variable auxiliar que nos dice cuál es la diferencia entre la demanda requerida y la cantidad de productos ya ensamblados,

se verifica si existen piezas suficientes para cubrir esta demanda y si es así entonces se ensamblan las piezas necesarias y se registra la cantidad en $piezasTerminadas$. Cuando se termina de verificar el proceso para los distintos productos se verifica cuáles piezas son las que lograron realizarse en la línea de ensamblaje de las cuales se hace un conteo junto con las piezas realizadas anteriormente.

El proceso termina después de que pasa una cantidad específica de periodos, y una vez que ha terminado el proceso se verifica cuál es la cantidad de productos de cada tipo que lograron ensamblarse por completo, lo cual se obtiene tomando la cantidad mínima de piezas tipo j del arreglo $piezasTerminadas_i$. Es decir, si para el producto tipo uno tenemos que necesita piezas tipo dos, tres y cinco y de cada una de ellas se logró ensamblar una demanda de veinte, quince y veinticinco productos respectivamente entonces la demanda total que pudo ensamblarse de ese producto uno sería de quince. Se suma el mínimo de la demanda de cada producto y el total de productos ensamblados es lo que tiene como respuesta nuestro algoritmo.

Al realizar este proceso se tiene un menor tiempo de espera en la línea de ensamblaje y la demanda puede realizarse parcialmente. Sin embargo, cuando las piezas que se han producido en la línea de producción no son suficientes para realizar el ensamblaje deseado por periodo entonces no realiza ningún ensamblaje, esta problemática se resuelve en el siguiente algoritmo.

4.1.2 Algoritmo de Ensambles Totales y Sub-ensambles

En el algoritmo 3, el cual es una heurística semi-voraz [13] ya que existe una lista de candidatos restringida que elige de manera aleatoria una cantidad limitada de candidatos para poder ingresar a la solución dependiendo de cuál es el mejor de ellos. En éste se consideran ensambles y sub-ensambles de los distintos productos i, como en el algoritmo anterior, esto se debe a que se considera que existen muchas estaciones de trabajo en la línea de ensamblaje, lo cual tiene como consecuencia que

Algoritmo 2 Ensambles de productos con demandas parcialmente terminadas

Input: *productos, cantPiezas, tipoPiezas, cantEnsamble, demanda, piezasPeriodo*

Output: cantidad de productos ensamblados (*prodTerminados*)

$p \leftarrow 0$
repeat
  $p \leftarrow p + 1$
  for $i \in \{1, ..., productos\}$ **do**
    if $piezasPeriodo_i^p[\alpha] \geq cantPiezas_i[\alpha]$, $\forall \alpha \in tipoPiezas_i$
      for $\alpha \in \{1, ..., \mid tipoPiezas_i \mid\}$ **do then**
        $prodEnsamble_i[\alpha] \leftarrow$ *floor(* $cantEnsamble_i[\alpha]$ */* $cantPiezas_i[\alpha]$ *)*
        if $demanda_i[\alpha] - piezasTerminadas_i[\alpha] \geq prodEnsamble_i[\alpha]$ **then**
          $a_i[\alpha] \leftarrow cantPiezas_i[\alpha] \times prodEnsamble_i[\alpha]$
          if $piezasPeriodo^p[\alpha] \geq a_i[\alpha]$ **then**
            $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $a_i[\alpha]$
            $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + prodEnsamble_i[\alpha]$
        else
          $aux \leftarrow demanda_i[\alpha] - piezasTerminadas_i[\alpha]$
          $b_i[\alpha] \leftarrow cantPiezas_i[\alpha] \times aux$
          if $piezasPeriodo^p[\alpha] \geq b_i[\alpha]$ **then**
            $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha] - b_i[\alpha]$
            $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + aux$

      end for
  end for
  $piezasPeriodo \leftarrow piezasPeriodo \cup piezasPeriodo^p$
until $p < periodosTotales$
for $i \in \{1, ..., productos\}$ **do**
  $prodTerminados \leftarrow prodTerminados + min(piezasTerminadas_i)$
end for
return *prodTerminados*

se verifique si se pueden hacer ensambles para todos los productos i en un solo periodo p. Sin embargo, cuando se considera una importancia de cubrimiento de demanda, es decir, cuando un producto tiene prioridad sobre otro, entonces puede suceder que el producto más importante no termine con su demanda por haber utilizado el tipo de piezas que necesita en ensamblar otros productos. Es por esa razón que durante los primeros periodos de este algoritmo, como se muestra de la línea siete a la diez, se considera que no puede comenzar a ensamblar la línea de ensamblaje si no existen piezas suficientes para realizar un ensamble total de cierta demanda del producto i, después de ciertos periodos entonces pueden comenzarse a hacer sub-ensambles de los productos, como se muestra de las líneas once a la trece.

Aún así, la problemática que se tiene en el algoritmo general es que a pesar de que existen demandas parciales y sub-ensambles de los productos, si no existen piezas suficientes para cubrir esa demanda parcial entonces no se realiza ningún ensamble.

Esta problemática puede resolverse considerando lo propuesto en el algoritmo 4, el cual forma parte del algoritmo 3. Para cada tipo de piezas se calcula cuál es la cantidad de productos demandados que se puede ensamblar, se guarda en la variable *prodEnsamble* (véase la línea dos) y se verifica si la diferencia entre la demanda requerida y los productos que ya han sido ensamblados con ese tipo de piezas supera el resultado de la variable *prodEnsamble*. Si es así entonces se verifica que la cantidad de piezas de tipo j que existen en ese periodo p sea suficiente para llevar a cabo el ensamble de los productos. Cuando la cantidad de piezas que se han realizado en la línea de producción sí es suficiente para ensamblar los productos que se tienen en la variable *prodEnsamble* entonces el algoritmo pasa al proceso en donde se ensamblan los productos y se hace un conteo de ellos, el resultado es guardado en la variable *piezasTerminadas*, como se muestra de la línea cuatro a la seis. En cambio, si las piezas que se realizaron en la línea de ensamblaje no son suficientes para poder llevar a cabo la demanda que se requiere en la variable *prodEnsamble* entonces se calcula el mayor entero menor que la división entre las piezas de ese tipo j que ya se han producido y la cantidad de piezas de ese tipo que el producto i requiere, lo cual es

Algoritmo 3 Ensambles y sub-ensambles de productos con demandas parcialmente terminadas

Input: *productos, cantPiezas, tipoPiezas, cantEnsamble, demanda, piezasPeriodo*

Output: productos ensamblados (*prodTerminados*)

1: $p \leftarrow 0$
2: **repeat**
3: $p \leftarrow p + 1$
4: **if** $p < (5/4)periodosTotales$ **then**
5: **for** $i \in \{1, ..., productos\}$ **do**
6: **if** $piezasPeriodo^p[\alpha] \geq cantPiezas_i[\alpha], \forall \alpha \in tipoPiezas_i$ **then**
7: *LineaEnsamblaje(periodosTotales, cantEnsamble, cantPiezas, prodTerminados, piezasTerminadas, piezasPeriodo)**
8: **end for**
9: **else**
10: **for** $i \in \{1, ..., productos\}$ **do**
11: *LineaEnsamblaje(periodosTotales, cantEnsamble, cantPiezas, prodTerminados, piezasTerminadas, piezasPeriodo)**
12: **end for**
13: $piezasPeriodo \leftarrow piezasPeriodo \cup piezasPeriodo^p$
14: **until** $p < 2 \times periodosTotales$
15: **for** $i \in \{1, ..., productos\}$ **do**
16: $prodTerminados \leftarrow prodTerminados + min(piezasTerminadas_i)$
17: **end for**
18: ***return*** *prodTerminados*

* LineaEnsamblaje es referente al Algoritmo 4

guardado en la variable *numEnsamble*. Se lleva a cabo solamente el ensamble de esa demanda y se hace conteo de los productos ensamblados, lo cual podemos ver de las líneas ocho a la diez.

Un ejemplo de esto es que se tengan que realizar ocho productos tipo uno. Para ello necesitamos de dieciséis piezas tipo tres, pero solamente tenemos siete piezas de ese tipo, en el algoritmo anterior no podríamos ensamblar ningún producto ya que no se tienen las piezas suficientes para cubrir la demanda que se desea. En este algoritmo se considera la cantidad de productos que pueden ensamblarse con respecto a la cantidad de piezas que existen de ese tipo, entonces para este ejemplo se tendría que se pueden ensamblar tres productos tipo uno con la cantidad de piezas tipo tres que se tiene. Para ello se considera el mayor entero menor que la división entre la cantidad de piezas tipo tres que se hicieron en la línea de producción y la cantidad de piezas que necesita el producto uno de ese tipo, lo cual sería siete entre dos y el mayor entero menor que esto nos da como resultado tres. Por esa razón tendremos que solamente pueden ensamblarse tres productos tipo uno y entonces la demanda que restaría por cubrir sería de cinco.

Cuando la diferencia entre la demanda y la cantidad de productos ensamblados no supera a la cantidad de productos de la variable *prodEnsamble*, entonces se toma una variable auxiliar en la que tengamos el conteo de la demanda que resta considerando los productos que ya han sido ensamblados. Una vez que tenemos la cantidad de demanda restante obtenida en la variable *numEnsamble* se verifica si hay piezas suficientes para que la línea de ensamblaje realice su trabajo, de ser así se ensambla la demanda requerida y se hace conteo de los productos ensamblados. De lo contrario, se considera el mayor entero menor que la división entre las piezas que existen disponibles del tipo j y las piezas que requiere el producto de ese tipo. El cálculo de esta división es tomado como la demanda que se podrá realizar del producto, se ensambla esa demanda y se realiza el conteo de los productos ensamblados, esto lo podemos encontrar de la línea doce a la diecinueve.

Algoritmo 4 LineaEnsamblaje(periodosTotales, cantEnsamble, cantPiezas, prodTerminados, piezasTerminadas, piezasPeriodo)

1: **for** $\alpha \in tipoPiezas_i$ **do**
2: $prodEnsamble_i[\alpha] \leftarrow floor(\ cantEnsamble_i[\alpha]\ /\ cantPiezas_i[\alpha]\ )$
3: **if** $demanda_i[\alpha] - piezasTerminadas_i[\alpha] \geq prodEnsamble_i[\alpha]$ **then**
4: $a_i[\alpha] \leftarrow cantPiezas_i[\alpha] \times prodEnsamble_i[\alpha]$
5: **if** $piezasPeriodo^p[\alpha] \geq cantPiezas_i[\alpha] \times prodEnsamble_i[\alpha]$ **then**
6: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $a_i[\alpha]$
7: $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + prodEnsamble_i[\alpha]$
8: **else**
9: $numEnsamble_i[\alpha] \leftarrow floor(\ piezasPeriodo^p[\alpha]\ /\ cantPiezas_i[\alpha]\ )$
10: $b_i[\alpha] \leftarrow numEnsamble_i[\alpha] \times cantPiezas_i[\alpha]$
11: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $b_i[\alpha]$
12: $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + numEnsamble_i[\alpha]$
13: **else**
14: $aux \leftarrow demanda_i[\alpha] - piezasTerminadas_i[\alpha]$
15: **if** $piezasPeriodo^p[\alpha] \geq aux \times cantPiezas_i[\alpha]$ **then**
16: $c_i[\alpha] \leftarrow cantPiezas_i[\alpha] \times aux$
17: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $c_i[\alpha]$
18: $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + aux$
19: **else**
20: $numEnsamble_i[\alpha] \leftarrow floor(\ piezasPeriodo^p[\alpha]\ /\ cantPiezas_i[\alpha]\ )$
21: $d_i[\alpha] \leftarrow numEnsamble_i[\alpha] \times cantPiezas_i[\alpha]$
22: $piezasPeriodo^p[\alpha] \leftarrow piezasPeriodo^p[\alpha]$ - $d_i[\alpha]$
23: $piezasTerminadas \leftarrow piezasTerminadas_i[\alpha] + numEnsamble_i[\alpha]$
24: **end for**

Volviendo al algoritmo 3, una vez que se lleva a cabo el proceso de ensamblar los productos se realiza un conteo de las piezas que surgen en la línea de producción en el periodo p en el que el proceso se encuentre. Esto se repite tantas veces como periodos se hayan propuesto y por último se busca la cantidad mínima de las demandas ensambladas por producto, esto nos dará la información sobre cuántos productos totales han sido ensamblados de cada tipo i, se suma la cantidad de demanda que se realizó de cada tipo y como respuesta obtenemos la cantidad total de productos ensamblados al final del proceso.

4.2 Algoritmo de Procesamiento de Información del Problema Producto-Pieza-Molde-Máquina

Para llevar a cabo la organización en la línea de ensamblaje se tienen instancias que se crearon con la información que tiene la línea de producción de cuántos productos son los que se deben realizar, cuántos tipos de pieza existen, cuántos moldes son los que hay disponibles y cuántas máquinas son las que están en funcionamiento.

Además, se tiene como información las cavidades que tiene cada molde para los distintos tipos de pieza, el tiempo que tarda un molde en hacer una cantidad de algún tipo de pieza, qué tipo de piezas necesita cada producto y cuántas piezas necesita de cada tipo, cuáles son los tiempos en que una máquina puede estar funcionando durante un periodo t, la importancia que tiene cada producto y la demanda de cada uno de ellos.

Como se muestra en el algoritmo 5, tenemos vectores para ver qué pasa con los moldes que están en funcionamiento, qué productos son los que se requieren, las piezas que se obtienen en cada periodo p y cómo están activas las máquinas en los periodos t. La información que se necesita obtener de las instancias del Problema

Producto-Pieza-Molde-Máquina es principalmente saber cuáles son los tipos de piezas que cada uno de los productos necesita y cuántas piezas necesita de cada tipo, la importancia de cada uno de estos productos y la demanda que se tiene de cada uno de ellos. Una vez que tenemos esta información de las instancias podemos realizar los cálculos y operaciones necesarias para crear las instancias que son necesarias para el algoritmo de la línea de ensamblaje.

Lo primero que el algoritmo realiza es tener en cuenta la información sobre el tiempo en que los moldes han estado activos en cada periodo t, es decir, cuánto tiempo estuvo cada uno de los moldes en las distintas máquinas en esos periodos, esto se guarda en el vector *moldes* como se muestra de las líneas cinco a la siete.

Luego, como se tiene ya la información del tiempo (en segundos) de lo que tarda un conjunto de piezas en producirse por cada tipo de pieza que hay, entonces se toma la información del tiempo; se calcula la cantidad de periodos p que necesita un montón de piezas tipo j en un molde k con una máquina l específica para realizarse. Esto se toma en cuenta teniendo el menor enterp mayor que la división entre la cantidad de segundos que tarda en hacerse un montón de piezas j y la cantidad en segundos que dura un periodo p, esto es, se toma el entero mayor más cercano del resultado de la división y esto se guarda en la variable $tiempoBatch$, como se muestra en la línea diez.

Una vez que se tiene la información de cuántos periodos se necesitan para el montón de piezas j entonces se verifica si la cantidad de periodos es igual a uno o no, si es igual a uno entonces se ve cuántos montones de piezas j pueden realizarse en ese periodo. Para ello se lleva a cabo el cálculo del mayo entero menor que la división entre el tiempo en segundos que dura un periodo p y el tiempo en segundos que tarda en hacerse un montón de piezas tipo j. Esto es, se toma el entero anterior más cercano del resultado de la división y se guarda en la variable x como se muestra en la línea doce, una vez que se tiene cuántos montones de pieza j se pueden realizar en un periodo p se calcula el total de periodos p que son necesarios para realizar

el total de montones de piezas j que el Problema Producto-Pieza-Molde-Máquina arrojó como resultado. Como se muestra en la línea trece, éste se guarda en la variable $batchTotales$, además se calcula la cantidad de piezas j que se obtienen en cada periodo p, esto se obtiene multiplicando la cantidad de montones que se pueden realizar en el periodo y la cantidad de cavidades que tiene el molde k que se está utilizando y se guarda en la variable $piezasBatch$, como se muestra en la línea catorce.

Cuando la variable $tiempoBatch$ no es igual a uno entonces se toma en cuenta que la variable $batchTotales$ tomará que la variable $batchTotales$ será exactamente la cantidad de montones totales del tipo j que arrojó como resultados el Problema Producto-Pieza-Molde-Máquina y la variable $piezasBatch$ tomará el total de cavidades que el molde k tiene, como se muestra en las líneas dieciséis y diecisiete. Una vez que se tienen todos estos datos se considera tambiéan cuál es el total de piezas que surgirán de cada tipo j considerando la multiplicación de la variable $batchTotales$ y la variable $piezasBatch$, esto se guarda en un vector para cada una de las piezas y se guarda la información del molde y la máquina en la que se están considerando que se realizaron.

Además, se tiene como información para las instancias la cantidad de piezas de cada tipo j que necesita cada producto i e indica el tipo de piezas j que requiere, esto puede verse de las líneas diecinueve a la veintitrés. Por último, se tiene cuál es la demanda de cada tipo de productos, la prioridad de ensamblaje que tiene cada producto y se guarda toda la información en la instancia, como se muestra de las líneas veinticuatro a veintinueve.

Dado a que los periodos p no son tomados por segundos sino por minutos se tiene que no todas las piezas que se tienen como respuesta del Problema Producto-Pieza-Molde-Máquina pueden lograr ser parte de las instancias creadas para llevar a cabo los algoritmos realizados de la línea de ensamblaje. Por ello se toma, de manera aleatoria, la cantidad necesaria para cubrir los tiempos propuestos en el Problema

Algoritmo 5 Instancias

Input: Respuesta del Problema Producto-Pieza-Molde-Máquina

Output: Instancia para algoritmo del Problema de Línea de Ensamblaje

for $i \in \{1, ..., t\}$ **do**
  while $y_{kl}^{t} = 1$ for $k \in \{1, ..., \mid moldes \mid\}$ and $l \in \{1, ..., \mid maquinas \mid\}$ **do**
    while $periodos > 0$ **do**
      $moldes \leftarrow moldes \cup y_{kl}^{t}$
for $j \in \{1, ..., piezas\}$ **do** for $k \in \{1, ..., \mid moldes \mid\}$ and $l \in \{1, ..., \mid maquinas \mid\}$
  if $j \in moldes$ **then**
    $tiempoBatch_{jkl}^{p} \leftarrow ceil(batchTime_{jkl}^{t}/tiempo)$
    if $tiempoBatch_{jkl}^{p} \leftarrow 1$ **then**
      $x_{jkl}^{p} \leftarrow floor(tiempo/batchTime_{jkl}^{t})$
      $batchTotales_{jkl}^{p} \leftarrow x_{jkl}^{t}/x_{jkl}^{p}$
      $piezasBatch_{j}^{p} \leftarrow x_{jkl}^{p} \times cavidades_{k}$
    else
      $batchTotales_{jkl}^{p} \leftarrow x_{jkl}^{t}$
      $piezasBatch_{j}^{p} \leftarrow cavidades_{k}$
  $totalPiezas \leftarrow totalPiezas_{jkl}^{p} \cup (batchTotales_{jkl}^{p} \times piezasBatch_{j}^{p})$
for $l \in \{1, ..., \mid maquinas \mid\}$ **do**
  for $p \in \{1, ..., periodosTotales * t\}$ **do**
    if $moldes_{jkl}^{p} \leftarrow 1 \ \forall j \in \{1, ..., piezas\}$ *and* $\forall k \in \{1, ..., \mid moldes \mid\}$ **then**
      $cantPiezas \leftarrow cantPiezas \cup totalPiezas_{jkl}^{p}$
      $tipoPiezas \leftarrow tipoPiezas \cup tipoPiezas_{jkl}^{p}$
for $i \in \{1, ..., productos\}$ **do**
  $demanda \leftarrow demanda \cup demanda_{i}$
  $prioridad \leftarrow prioridad \cup prioridad_{i}$
for $i \in \{1, ..., p \times t\}$ **do**
  $piezasPeriodo \cup p \cup tipoPiezas_{i} \cup cantPiezas_{i}$

Producto-Pieza-Molde-Máquina y el resto de las piezas no se toma en cuenta para la instancia.

En el siguiente capítulo se muestran los experimentos que se realizaron para la comprobación del funcionamiento de los algoritmos propuestos.

Capítulo 5

Resultados Experimentales

Cada uno de los algoritmos propuestos en el capítulo anterior fue probado para determinar si logran obtener un resultado deseable. El algoritmo 1 fue utilizado para experimentación de instancias pequeñas que fueron útiles para comprender el Problema de Línea de Ensamblaje, es por ello que en esta sección solamente se muestran resultados de los algoritmos 2 y 3 los cuales consideran las instancias generadas por el algoritmo 5. Estos algoritmos fueron realizados y ejecutados en el lenguaje de programación `Python` versión 3.6.4.

Para realizar los experimentos de los algoritmos anteriores se seleccionaron instancias del Problema Producto-Pieza-Molde-Máquina que tienen como características una cantidad de cincuenta productos distintos, cincuenta tipos de piezas diferentes, diez máquinas que están en funcionamiento y el requerimiento de diez, veinte y treinta moldes para poder realizar las piezas necesarias indicadas. Se llevaron a cabo los cálculos necesarios para realizar la unión entre la línea de ensamblaje y la línea de producción y la obtención de información de las instancias del Problema Producto-Pieza-Molde-Máquina procesadas en el lenguaje de programación `R` versión 3.4.3. Se seleccionaron ocho tipos de instancias para cuando están en funcionamiento diez moldes, ocho tipos de instancias para cuando funcionan veinte moldes y otros ocho tipos para cuando se tienen en funcionamiento treinta moldes. Además, se consideraron dos diferentes periodos t de la línea de producción, los cuales fueron valores

de tres para cuando la cantidad de moldes en funcionamiento era de diez y veinte, y de valor dos para cuando la cantidad de moldes en funcionamiento fuera de treinta. Estos experimentos fueron ejecutados en una computadora iMac 8.1 con procesador Intel Core 2 Duo, velocidad 3.06 GHz y memoria de 4GB.

El objetivo de nuestro trabajo es la asignación de tiempos en la organización de la línea de ensamblaje, es por ello que el principal estudio de los resultados es la cantidad de periodos que la línea de ensamblaje requiere para lograr realizar la mayor cantidad de productos ensamblados durante ciertos periodos p. Sin embargo, también es de nuestro interés verificar la cantidad de productos que lograron ensamblarse por completo ya que esto nos dice qué tan bueno es el algoritmo y además la cantidad de piezas que se quedaron en el almacén, lo cual es perjudicial para nuestra línea de ensamblaje al generarse costos de almacenamiento.

Para cada una de las instancias se llevaron a cabo cinco corridas ya que se tiene una lista de candidatos restringida, la cual nos indica cuáles tipos de productos son los que están disponibles en un cierto periodo p para su ensamblaje, esto es debido a que no existen estaciones de trabajo disponibles suficientes para realizar el ensamblaje de todos los distintos productos que son demandados.

5.1 Periodos p

Cuando se trata de asignar tiempos en una línea de ensamblaje se hace referencia sobre minimizar los tiempos de ocio entre un producto ensamblado y otro. Conforme se realizaron avances en la investigación y en los algoritmos que se llevaron a cabo para poder lograr esta asignación se hizo notorio que, con respecto a los resultados del total de periodos que requiere el algoritmo para completar los productos finales, el algoritmo 3 la mayoría de las veces es mejor que el algoritmo 2.

En la figura 5.1 se muestran las gráficas que se realizaron con los resultados obtenidos de ambos algoritmos, para cada uno de los casos (a), (b) y (c) se tiene que

en el eje llamado *Instancias* se consideran las instancias generadas por el algoritmo 5 y en el eje llamado *Periodospromedio* se considera la cantidad de periodos p que tomó la Línea de Ensamblaje para terminar el proceso de ensamblaje de productos. Para poder obtener estos resultados se realizaron cinco corridas de cada tipo de instancia obtenida a partir de la solución del Problema de Producto-Pieza-Molde-Máquina dado que no siempre se tiene la misma respuesta debido a la aleatoriedad del algoritmo.

Una vez teniendo los resultados del algoritmo se calcularon promedios para cada una de las distintas instancias, es decir, para cada tipo de instancia se obtuvo un promedio con las cinco corridas en las que se utilizó esa instancia, así se logró obtener un aproximado del resultado que se obtendría conforme a las instancias elegidas.

En el inciso (a) se puede observar la gráfica que nos muestra los resultados para aquellas instancias que tienen como características el tomar un periodo $t = 3$ en la línea de producción. Esto es, que se está considerando un tiempo de tres semanas en la línea de producción, y que la cantidad de moldes que se pueden estar en funcionamiento es de diez, además de que se consideran cincuenta tipos de productos, cincuenta tipos de piezas y diez máquinas en funcionamiento. Se tienen las cantidades de periodos p promedio que cada uno de los algoritmos requiere para realizar su proceso, el cual no deja de realizarse hasta que ya no haya piezas por ensamblar o se llegue a un determinado número de periodos establecido. Para el algoritmo 2 se tienen los resultados en color azul y para el algoritmo 3 se muestran en color rojo. Se observa que en la mayor parte de las instancias el algoritmo 3 requiere una cantidad menor o igual de periodos para realizar su proceso que el algoritmo 2, a excepción de una instancia en la cual el algoritmo 3 es mayor, así que se puede decir que el algoritmo 3 es mejor que el algoritmo 2.

En el inciso (b) se tienen aquellas instancias en las que se tiene un periodo $t = 3$ de la línea de producción, en donde se consideran cincuenta tipos distintos de

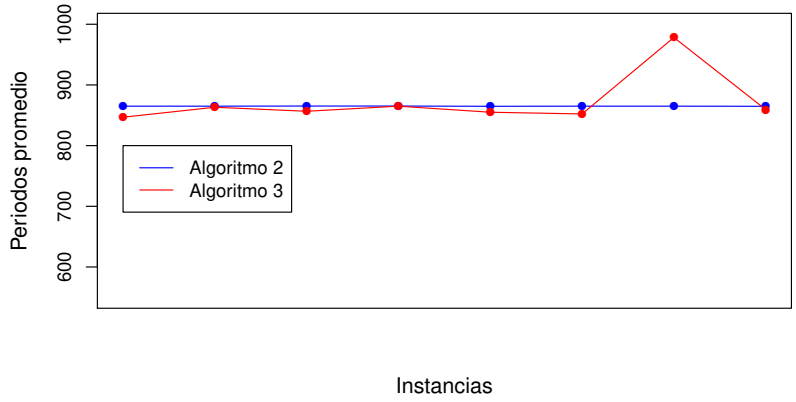
productos, cincuenta tipos distintos de piezas, diez máquinas en funcionamiento y veinte moldes distintos que pueden ser utilizados en las máquinas. Se puede observar que entre los algoritmos 2 y 3 existe una similitud de periodos promedio para ciertas instancias, en dos ocasiones el algoritmo 2 es mejor que el algoritmo 3 y en una ocasión el algoritmo 3 es mejor que el algoritmo 3, sin embargo, cuando el algoritmo 2 es peor que el algoritmo 3 existe una diferencia mayor que cuando el algoritmo 3 es peor que el algoritmo 2.

Por último, en el inciso (c) se muestran los resultados de aquellas instancias en las cuales se tiene un periodo $t = 2$ en la línea de producción, una cantidad de cincuenta productos distintos, diez máquinas en funcionamiento, cincuenta piezas distintas y treinta moldes que están disponibles. En cuanto a estos parámetros, se tiene que en la mayoría de las instancias el algoritmo 2 es mejor que el algoritmo 3, aun cuando en dos ocasiones ambos algoritmos tienen el mismo tiempo promedio y en una ocasión el algoritmo 3 es mejor que el algoritmo 2.

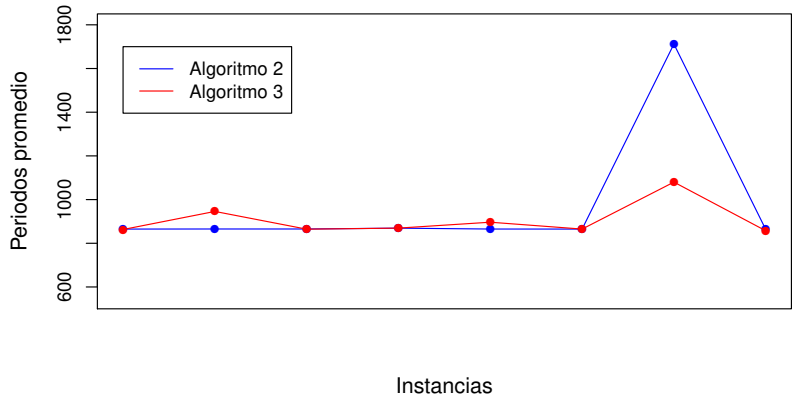
Esta mejoría se debe a que en el algoritmo 3 se consideran sub-ensambles de productos, de esta manera el proceso se hace más rápido ya que se tiene que los productos van quedando en espera si es que no se terminaron de ensamblar y en cuanto llegan nuevas piezas se ensamblan totalmente, en cambio, en el algoritmo 2 se dejan en espera los productos hasta que se tienen todas las piezas disponibles y esto retrasa el proceso de ensamblaje.

5.2 Productos Ensamblados

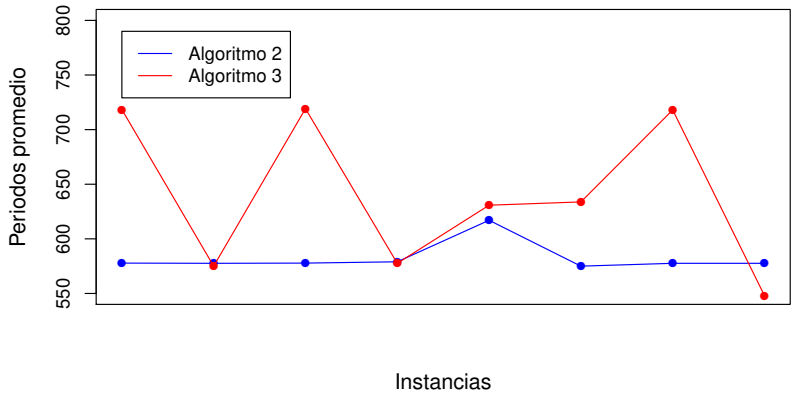
A pesar de que el objetivo de esta investigación es asignar tiempos para la organización de la línea de ensamblaje no sería suficiente con verificar los periodos p que ésta necesita para realizar su proceso, sino que se debe tomar en cuenta también que la organización de los tiempos sea la adecuada para que la mayor cantidad de productos finales surjan al finalizar el proceso, es por eso que se considera la cantidad



(a) Periodos $t = 3$, Moldes $= 10$



(b) Periodos $t = 3$, Moldes $= 20$



(c) Periodos $t = 2$, Moldes $= 30$

Figura 5.1: Resultados de los periodos promedio de los algoritmos 2 y 3.

de productos finales que surgen cuando el proceso de ambos algoritmos termina.

Para ello se realizaron cinco corridas de cada una de las instancias, en las cuales se tomaron en cuenta como parámetros de $t = 3$ para aquellas instancias que tienen diez y veinte moldes disponibles que pueden utilizarse en las distintas máquinas de acuerdo a la compatibilidad que exista entre ellos y de $t = 2$ para aquellas instancias que tienen disponibles treinta moldes. Una vez realizadas las corridas se tomaron para graficar diagramas de caja que nos ayuden a ver si existe diferencia entre una corrida y otra y entre las distintas instancias tomadas para cada algoritmo, como se muestra en la figura 5.2, en la cual se puede observar la cantidad de productos ensamblados finales que surgen en los algoritmos 2 y 3 con respecto a las instancias obtenidas que tienen las características mencionadas anteriormente.

Para cada uno de los casos (a), (b) y (c) se tiene que en el eje llamado *Instancias* se consideran las instancias generadas por el algoritmo 5 y en el eje llamado $Productos_Ensamblados$ se considera la cantidad de productos i totales que lograron obtener su ensamblaje total. En el inciso (a) se tienen aquellas instancias que cuentan con cincuenta distintos productos, cincuenta tipos de piezas diferentes, periodos de $t = 3$ en la línea de producción, diez máquinas en funcionamiento y diez moldes distintos disponibles. De acuerdo con la gráfica se puede observar que en la mayoría de las instancias la cantidad de productos ensamblados es mayor en el algoritmo 3 que en el algoritmo 2, incluso en la mayoría de las instancias el algoritmo 2 no termina de hacer ningún producto. Además, no existe mucha diferencia entre una corrida y otra por cada instancia, sin embargo, existen algunos datos atípicos que no son tan lejanos a la media de éstas. En la mayoría de las instancias el algoritmo 3 realiza una muy buena cantidad de productos, excepto en la instancia cinco, en donde no se obtuvieron muchos productos finales para ninguno de los dos algoritmos.

En el inciso (b) se tienen aquellas instancias que cumplen con las características de tener cincuenta tipos de productos distintos, cincuenta tipos de piezas diferentes, periodos $t = 3$ en la línea de producción, diez máquinas que están en funciona-

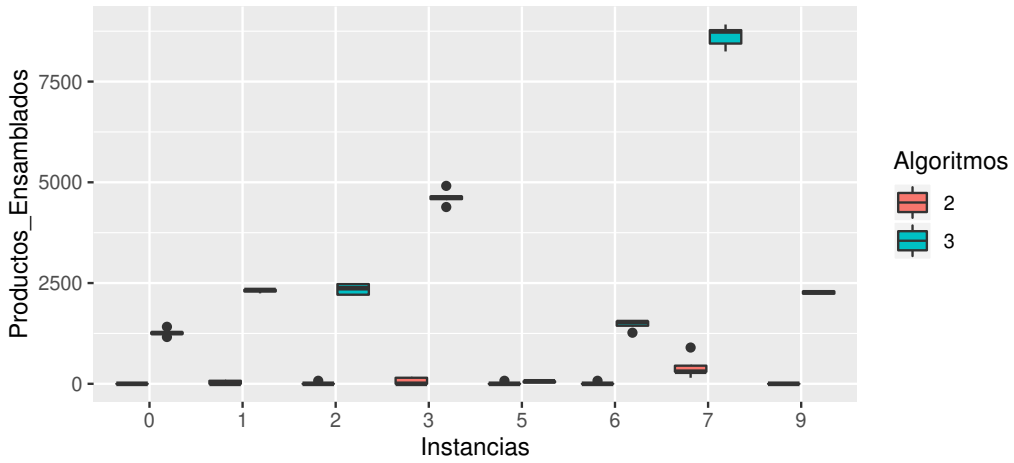
miento y veinte moldes distintos que pueden ser montados en las máquinas. Para las instancias con estas características tenemos como resultados que el algoritmo 3 es mejor que el algoritmo 2 ya que la cantidad de productos finalizados es mayor, excepto cuando se considera la instancia número nueve. La diferencia que hay con la gráfica anterior es que con estas instancias que tienen veinte moldes disponibles existe mayor variedad para poder elaborar las piezas en la línea de producción, de esta manera pueden realizarse más productos finales en ambos algoritmos y para el algoritmo 2 existen más instancias que realizan productos finales a diferencia de las instancias anteriores y el algoritmo 3 supera la cantidad de productos ensamblados a sí mismo con los resultados del inciso anterior.

En el inciso (c) se tienen las instancias que cuentan con las características de cincuenta productos, cincuenta tipos de piezas diferentes, periodos $t = 2$ en la línea de producción, diez máquinas en funcionamiento y treinta moldes que están disponibles. Para estas instancias se tiene como resultado que en la mayoría de los casos el algoritmo 3 existe mayor cantidad de productos finales en comparación con el algoritmo 2 y éste tiene mayor cantidad de productos finalizados en comparación con las instancias con las otras características, sin embargo existen dos instancias en las que no se termina ningún producto en ambos algoritmos.

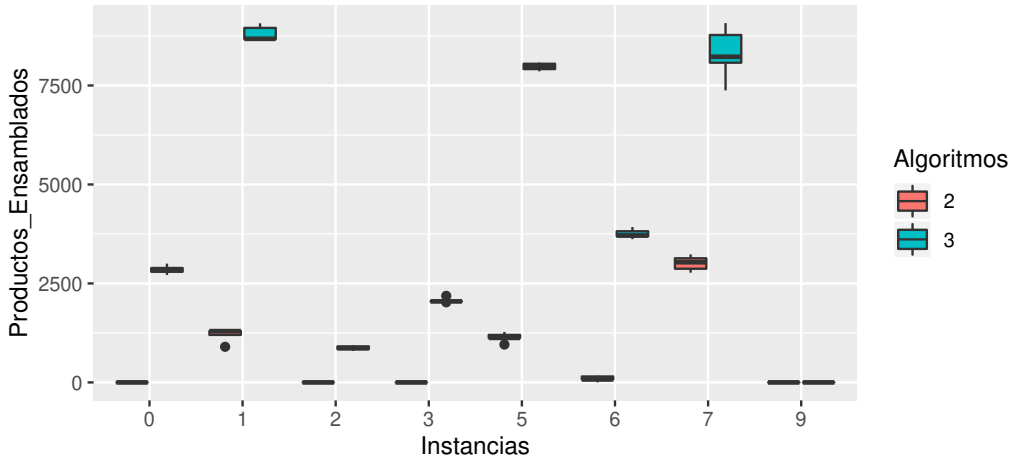
La mejora en el algoritmo 3 sobre la gran cantidad de productos finales a diferencia del algoritmo 2 se debe a que se considera hacer ensambles y sub-ensambles para los productos de mayor importancia pero si aún no existen piezas suficientes para los más importantes se considera ensamblar y sub-ensamblar aquellos que le siguen.

5.3 Piezas Almacenadas

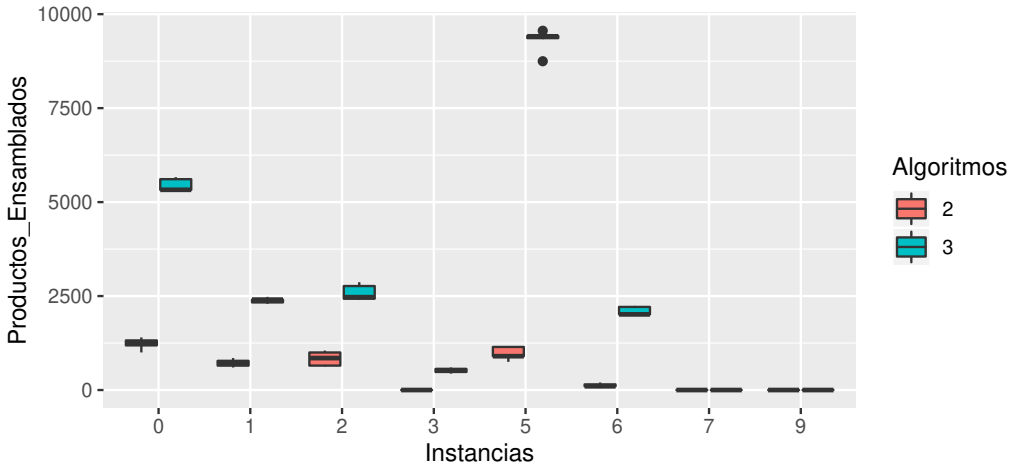
Cuando se habla sobre la disminución de periodos en la línea de ensamblaje también se habla de que la mejor manera de lograr el objetivo de obtener la mayor



(a) Periodos $t = 3$, Moldes $= 10$



(b) Periodos $t = 3$, Moldes $= 20$



(c) Periodos $t = 2$, Moldes $= 30$

Figura 5.2: Resultados de los productos ensamblados de los algoritmos 2 y 3.

cantidad de productos finales es que exista la menor cantidad de piezas que se queden en el almacenamiento. Es por eso que se realizaron cinco corridas para cada una de las instancias seleccionadas de la línea de producción y se realizaron gráficas de diagramas de caja que muestran el resultado de estos experimentos, como se ve en la figura 5.3.

Para cada uno de los casos (a), (b) y (c) se tiene que en el eje llamado *Instancias* se consideran las instancias generadas por el algoritmo 5 y en el eje llamado $Piezas_Almacenadas$ se considera la cantidad piezas tipo j que fueron almacenadas al finalizar el proceso de ensamblaje. En el inciso (a) se muestran los resultados de las instancias que tienen como características cincuenta productos distintos, cincuenta tipos de piezas diferentes, periodos $t = 3$ de la línea de producción, diez moldes disponibles y diez máquinas en funcionamiento. Como se puede observar, en la mayoría de las instancias la cantidad de piezas que se encuentran en el almacenamiento es de más de mil para el algoritmo 2, mientras que para el algoritmo 3 se tiene una cantidad de cero piezas que se quedan para el almacenamiento. Si la idea es que exista una mayor cantidad de productos que puedan ser ensamblados en su totalidad y se tiene una gran cantidad de piezas en el almacenaje cuando cada uno de los distintos productos no utiliza más de cinco piezas de algún tipo entonces se puede decir que más de mil piezas en almacenamiento es una gran cantidad de piezas que podrían ser utilizadas en alguno de los productos, además no existe una gran diferencia entre una corrida y otra para cada una de las instancias utilizadas. En el caso del algoritmo 3 sucede que existen sub-ensambles que permiten que algún producto esté en un porcentaje ensamblado de su totalidad, es por eso que la cantidad de piezas que quedan sin ensamblar es de cero.

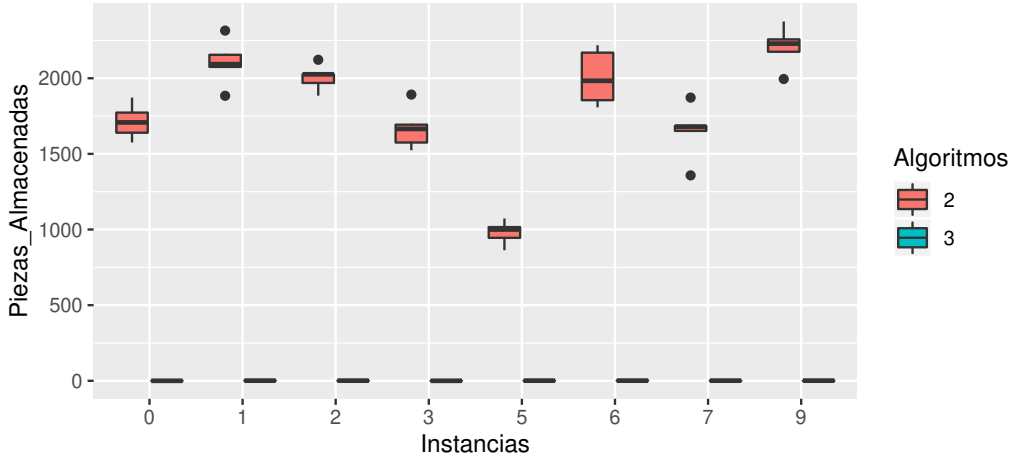
En el inciso (b) se tienen los resultados de aquellas instancias que cuentan con las características de considerar cincuenta productos distintos, cincuenta tipos de piezas diferentes, veinte moldes disponibles, periodos $t = 3$ en la línea de producción y diez máquinas en funcionamiento. Dado que existen más moldes que pueden ser utilizados, la cantidad de piezas que quedan almacenadas en el algoritmo 2 disminuye

con respecto a los resultados mostrados en el inciso (a), sin embargo este algoritmo sigue teniendo muchas piezas que quedan en el almacenamiento, lo cual no es bueno para la línea de ensamblaje ya que ésta intenta utilizar cada pieza para realizar los productos correspondientes a la demanda requerida. Se puede observar que el algoritmo 3 continúa teniendo cero piezas que se van a almacenaje en todas las instancias seleccionadas.

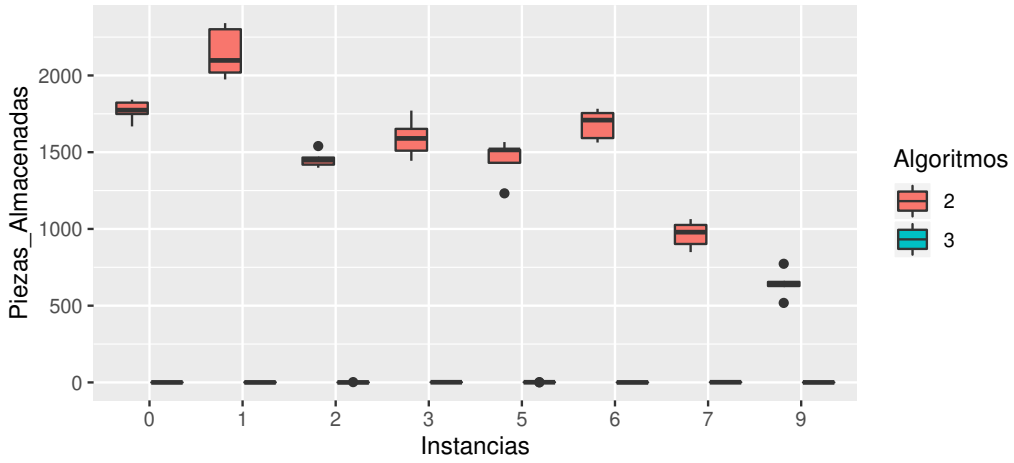
En el inciso (c) se tienen las instancias que cuentan con las características de periodos $t = 2$ en la línea de producción, cincuenta tipos de productos demandados, cincuenta tipos de piezas distintas, treinta moldes que están disponibles y diez máquinas que están en funcionamiento. En los resultados que se pueden observar sobre la cantidad de piezas almacenadas tenemos que en todas las instancias el algoritmo 2 cuenta con una gran cantidad de piezas en el almacenamiento, una cantidad mucho mayor que la cantidad de piezas almacenadas que resultan del algoritmo 3, aún cuando éste último tiene más de cero piezas en el almacenaje para algunas de las instancias.

Dado que se tiene que para el algoritmo 3 existe una mayor cantidad de productos finales en comparación con el algoritmo 2, existe una mejoría bastante notable en la cantidad de piezas que quedaron en el almacenamiento al finalizar el proceso en la línea de ensamblaje.

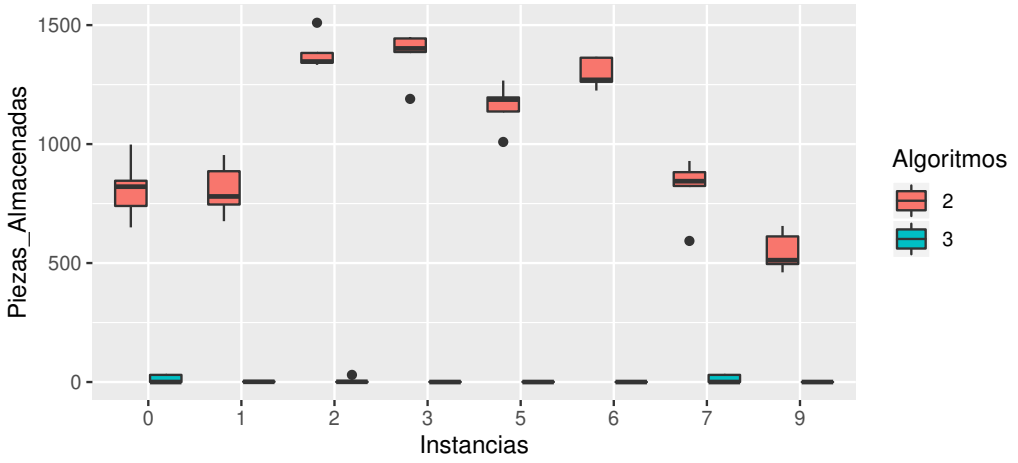
En el capítulo siguiente se muestran las conclusiones y trabajo a futuro que surgen del análisis de las respuestas a los algoritmos propuestos en el capítulo cuatro.



(a) Periodos $t = 3$, Moldes $= 10$



(b) Periodos $t = 3$, Moldes $= 20$



(c) Periodos $t = 2$, Moldes $= 30$

Figura 5.3: Resultados de las piezas en almacenamiento de los algoritmos 2 y 3.

Capítulo 6

Conclusiones y Trabajo a Futuro

Uno de los principales problemas en la línea de ensamblaje es que se tiene una gran espera en las estaciones de trabajo para que pueda comenzar a realizarse el proceso que se requiere, lo cual es debido a que no existe una comunicación a la par de la línea de ensamblaje con la línea de producción. Es por eso que se realizó el estudio del emparejamiento de ambas líneas para que logren tener un acercamiento a lo que sucede en tiempo real.

Es muy frecuente que al estudiarse una línea de producción se tenga como respuesta el lote obtenido de las piezas requeridas para ensamblar los productos demandados (que es la suma de lo montones de piezas que van surgiendo cada vez que un molde termina de realizar su proceso) que se obtiene al final de un tiempo establecido, el cual se considera comúnmente por horas, días, semanas o meses. Esto se debe a que se realizan todas las asignaciones y organizaciones posibles para que se obtenga al final de estos periodos la cantidad necesaria para lograr obtener los productos finales demandados. Sin embargo, en las investigaciones que existen se muestra como respuesta los lotes que se obtienen y no lo que va sucediendo con la línea de producción por cada montón que va surgiendo de las piezas en periodos de tiempo menores a los establecidos para dar información sobre los lotes. Es por esto

que la línea de ensamblaje trabaja con lo que va saliendo por lotes en la línea de producción, lo cual no resulta muy bien ya que se analizan los procesos de ensamblado de acuerdo a los tiempos de salida de los lotes y si lo que busca la línea de producción es disminuir los tiempos de ocio entre un ensamblaje y otro sería muy poco conveniente que existieran organizaciones con respecto a los periodos largos de tiempo que maneja la línea de producción. Si los tiempos en la línea de ensamblaje pudieran reducirse a los tiempos que se tienen en la línea de producción considerando los montones que surgen de los distintos tipos de piezas sería mejor que si se consideran los lotes.

6.1 Conclusiones

En este trabajo se presentaron algoritmos que llevan a cabo una asignación de tiempos en la organización de una línea de ensamblaje, los cuales consideran tomar la información procesada de la respuesta que surge en la línea de producción del Problema Producto-Pieza-Molde-Máquina. Una vez que se toma la información sobre en qué momentos surgen las cantidades de los distintos tipos de piezas, la cantidad de piezas y el tipo de piezas que se produce se considera a la par verificar si existe la posibilidad de que se comience a ensamblar algún producto que requiere las piezas que han surgido de la línea de ensamblaje en ese momento.

Se estudiaron los distintos tipos de algoritmos que se muestran con anterioridad, sin embargo se decidió que el algoritmo 2 y el algoritmo 3 son los algoritmos que sirvieron de apoyo para optimizar el Problema de Línea de Ensamblaje con todas las restricciones que se consideraron. En estos algoritmos se presenta la asignación de tiempos de la línea de ensamblaje, como respuesta se da el periodo de tiempo en el que se debe comenzar a hacer el ensamblaje de algún producto, el producto que puede ensamblarse, la cantidad de productos que pueden ensamblarse en ese periodo de tiempo y el tipo y cantidad de piezas que se requieren para realizar los productos requeridos.

Además, se tiene como respuesta a la hipótesis planteada la cantidad de periodos totales que la línea de ensamblaje requiere para terminar los productos demandados. Como dato adicional se considera las piezas que quedaron sin utilizarse ya que éstas deben quedar en almacenamiento y entre menos piezas haya almacenadas disminuye el costo de la producción. También se tiene como dato adicional la cantidad de productos que lograron finalizarse, ya que estos son los que pueden ir saliendo al mercado. La diferencia que existe entre un algoritmo y otro es que en el algoritmo 2 se tiene que pueden realizarse demandas parciales de los productos, sin embargo no existen sub-ensambles de los distintos productos; en cambio, en el algoritmo 3 existen demandas parciales de los productos y además sub-ensambles de ellos, los cuales pueden quedar pendientes mientras se hacen sub-ensambles de otros productos.

Se analizaron las respuestas de los algoritmos en base a distintas instancias que se realizaron al procesar la información obtenida del Problema Producto-Pieza-Molde-Máquina. Una vez obtenidos los resultados al realizarse repeticiones de cada una de las instancias en cada uno de los dos algoritmos elegidos (algoritmos 2 y 3) se generaron las gráficas con la información obtenida, las cuales son apoyo visual para analizar cada uno de estos algoritmos. En el caso en el que se analiza la totalidad de periodos de tiempo que requiere la línea de ensamblaje para terminar su trabajo se observa que ambos algoritmos tienen una similitud en cuanto a cantidades de tiempo requeridas, lo cual nos dice que no afecta en mucho qué algoritmo sea elegido, el proceso terminará a buen ritmo.

En el caso en el que tenemos el análisis de los productos que lograron ensamblarse por completo se puede observar que aquellos resultados que fueron mejores fueron los del algoritmo 3 para gran parte de las instancias seleccionadas y con una diferencia notoria en cuanto al algoritmo 2, ya que en este en muchas ocasiones la cantidad de productos finales fue de cero. En el caso en el que se analiza las piezas que quedan en el almacenamiento se tiene que el algoritmo 2 tiene muchas piezas que se deben almacenar, en cambio el algoritmo 3 en la mayoría de las ocasiones

resulta con cero piezas que deben ir a almacenamiento.

El análisis de información sobre el tiempo total promedio que tarda el proceso de la línea de ensamblaje en terminar su trabajo, la cantidad de piezas que quedaron en almacenamiento y la cantidad de productos que lograron ensamblarse por completo nos deja en claro que el algoritmo 3 es el indicado para llevar a cabo el proceso de la asignación de tiempos de una línea de ensamblaje y además con resultados muy buenos que ayudan a que no existan costos adicionales y que el trabajo se realice casi por completo.

6.2 Trabajo a futuro

En el presente trabajo se desarrollaron algoritmos que consideran una cantidad establecida de estaciones, una cantidad establecida de cantidad de piezas que pueden ensamblarse y se considera que solamente hay una persona por estación de trabajo. Una propuesta para mejorar la asignación de tiempos en la organización de la línea de ensamblaje es que se tome en cuenta de manera realista lo que existe en una línea de producción en cuanto a la cantidad de piezas que una persona puede ensamblar, la cantidad de personas que estén en cada estación y la cantidad de estaciones que existen en la línea de ensamblaje.

Otra de las propuestas es que no en todas las estaciones de trabajo se considere que pueden realizarse todos los ensamblajes requeridos, esto es, que existan estaciones que realicen exclusivamente el ensamblaje de algunos tipos de piezas o que existan estaciones que solamente realicen el ensamblaje de algunos tipos de productos para que exista un límite mayor de trabajo.

También se tiene en consideración que el tiempo en el que se verifica si la línea de producción ya ha hecho un montón de piezas de algún tipo sea considerado en pocos minutos e incluso segundos, ya que los moldes que están disponibles tienen distintos tiempos en los que realizan las piezas incluso si el tipo de pieza entre un

molde y otro es la misma.

Además se tiene en cuenta que pueda proporcionarse el mismo tiempo límite de una jornada laboral de la línea de producción en la línea de ensamblaje, ya que en esta investigación no existe un tiempo límite de periodos que puede considerar la línea de ensamblaje para terminar el trabajo demandado.

Dado que los algoritmos propuestos son heurísticas que sirven de apoyo para que los resultados puedan tenerse de manera rápida pero no consiguen una respuesta óptima se considera que se realice una metaheurística que sea de apoyo para poder mejorar la respuesta que se obtiene de la línea de ensamblaje, ya que dentro de la heurística que se desarrolló por ahora existe una lista de candidatos restringida la cual hace que cada vez que el algoritmo se realiza existe un cambio en su respuesta. Una de las metaheurísticas que se desean analizar es la de Procedimiento Voraz Aleatorio de Búsqueda Adaptativa (GRASP por sus siglas en inglés) dado que ya se analizó una heurística semi-voraz.

Bibliografía

[1] Joaquín Bautista, Elena Fernández, J L González Velarde, and Manuel Laguna. Hiperheurística para un problema de equilibrado de líneas de montaje usando scatter search. In *Actas del IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB2005)*, pages 839–845, Granada, España, 2005.

[2] Ilker Baybars. A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32(8):909–932, 1986.

[3] Patrizia Beraldi, Gianpaolo Ghiani, Antonio Grieco, and Emanuela Guerriero. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, 35(11):3644–3656, 2008.

[4] Edward H Bowman. Assembly-line balancing by linear programming. *Operations Research*, 8(3):385–389, 1960.

[5] Nils Boysen, Malte Fliedner, and Armin Scholl. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111(2):509–528, 2008.

[6] GM Buxey. Assembly line balancing with multiple stations. *Management Science*, 20(6):1010–1021, 1974.

[7] Medina Chacón and Emilsy Rosio. Método heurístico para el balance de líneas de ensamble con consideraciones ergonómicas. *Ingeniería Industrial*, 2014.

[8] Oscar Leonel Chacón Mondragón, Omar Jorge Ibarra Rojas, Yasmín Águeda Ríos Solís, and Mario Alberto Saucedo Espinosa. Programación pieza-molde-máquina en planeación de la producción. *Ciencia UANL*, 15(58):59–65, 2012.

[9] Manuel Chica, Oscar Cordon, Sergio Damas, and Joaquin Bautista. Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: Aco and random greedy search. *Information Sciences*, 180(18):3465–3487, 2010.

[10] Satyaki Ghosh Dastidar and Rakesh Nagi. Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. *Computers & Operations Research*, 32(11):2987–3005, 2005.

[11] El Comercio. Mira en 2 minutos cómo se fabrica tu tarjeta madre. https://elcomercio.pe/tecnologia/actualidad/mira-2-minutos-fabrica-tarjeta-madre-video-387898, 2015.

[12] Knut Haase and Alf Kimms. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, 66(2):159–169, 2000.

[13] J Pirie Hart and Andrew W Shogan. Semi-greedy heuristics: An empirical study. *Operations Research Letters*, 6(3):107–114, 1987.

[14] Omar J Ibarra-Rojas, Roger Z Ríos-Mercado, Yasmin A Ríos-Solís, and Mario A Saucedo-Espinosa. A decomposition approach for the piece–mold–machine manufacturing problem. *International Journal of Production Economics*, 134(1):255–261, 2011.

[15] Omar J. Ibarra-Rojas, Yasmín A. Ríos-Solís, Marta Cabo, and Edgar Possani. Heuristic based on mathematical programming for a lot-sizing and scheduling problem in the mold-injection production. Reporte técnico PISIS–2017–06, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, 2017.

[16] Andres Jaramillo Garzon and Jorge Hernan Restrepo Correa. Aplicación de la programación dinámica para resolver el problema simple de balanceo de una línea de ensamble. *Scientia et Technica*, 3(46):62–67, 2010.

[17] Sophie D Lapierre, Angel Ruiz, and Patrick Soriano. Balancing assembly lines with tabu search. *European Journal of Operational Research*, 168(3):826–837, 2006.

[18] CKY Lin, CL Wong, and YC Yeung. Heuristic approaches for a scheduling problem in the plastic molding department of an audio company. *Journal of Heuristics*, 8(5):515–540, 2002.

[19] Herbert Meyr and Matthias Mann. A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. *European Journal of Operational Research*, 229(3):718–731, 2013.

[20] Raquel Murillo Garcia, Fabian Peñaherrera-Larenas, Ely Borja Salinas, and Valentino Vanegas. Líneas de ensamble y balanceo y su impacto en la productividad de los procesos de manufactura. *Revista Observatorio de la Economía Latinoamericana*, 2018.

[21] Nagen Nagarur, Prem Vrat, and Wanchai Duongsuwan. Production planning and scheduling for injection moulding of pipe fittings a case study. *International Journal of Production Economics*, 53(2):157–170, 1997.

[22] Seetharama L Narasimhan, Dennis W McLeavey, and Peter J Billington. *Planeación de la producción y control de inventarios*. Prentice-Hall Hispanoamericana, México, 1996.

[23] Eliana Mirledy Toro Ocampo, Mauricio Granada Echeverry, and Rubén Romero. Algoritmo memético aplicado a la solución del problema de asignación generalizada. *Tecnura*, 8(16):55–63, 2005.

[24] Diego León Peña Orozco, Ángela María Neira García, and Reynel Alberto Ruiz Grisales. Aplicación de técnicas de balanceo de línea para equilibrar las

cargas de trabajo en el área de almacenaje de una bodega de almacenamiento. *Scientia et Technica*, 21(3):239–247, 2016.

[25] Jorge Hernan Restrepo, Pedro Daniel Medina, and Eduardo Arturo Cruz. Problemas de balanceo de línea salbp-1 y salbp-2: un caso de estudio. *Scientia et Technica*, 14(40):105–110, 2008.

[26] L. C. Riascos Alvarez. *Formulations and Algorithms for the Kidney Exchange Problem*. Master thesis, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, NL, Mexico, April 2017.

[27] Yasmín A. Ríos Solís, Omar J. Ibarra Rojas, Marta Cabo, and Edgar Possani. A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production. Technical Report PISIS-2019-03, Universidad Autónoma de Nuevo León, 2019.

[28] Armin Scholl and Christian Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3):666–693, 2006.

[29] Armin Scholl, Nils Boysen, and Malte Fliedner. The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics. *OR Spectrum*, 35(1):291–320, 2013.

[30] Armin Scholl and Stefan Voß. Simple assembly line balancing—heuristic approaches. *Journal of Heuristics*, 2(3):217–244, 1997.

[31] T. Stützle and R. Ruiz. Iterated greedy. Reporte técnico TR/IRIDIA/2018-006, Université Libre de Bruxelles, Bruxelles, Belgium, 2018.

[32] Candace Arai Yano and Ahmet Bolat. Survey, development, and applications of algorithms for sequencing paced assembly lines. *Journal of Manufacturing and Operations Management*, 1989.

Resumen autobiográfico

Beatriz Alejandra García Ramos

Candidato para obtener el grado de
Maestría en Ciencias en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:
Problema de asignación de tiempos en la organización de una línea de ensamblaje

Nací el 20 de enero de 1995 en el municipio de San Nicolás de los Garza en el estado de Nuevo León. Mis padres Jesús García Gámez y Beatriz Ramos Larralde me han cuidado y educado desde mi nacimiento, al igual que a mi hermana Karina Guadalupe García Ramos. Concluí mis estudios como Licenciado en Matemáticas en junio del año 2017 en la Facultad de Ciencias Físico Matemáticas perteneciente a la Universidad Autónoma de Nuevo León. En agosto de 2017 ingresé a la Maestría en Ciencias en Ingeniería de Sistemas, la cual está a cargo del Posgrado de la Facultad de Ingeniería Mecánica y Eléctrica perteneciente a la Universidad Autónoma de Nuevo León.