CULCYT. Cultura Científica y Tecnológica
Vol. 23 | N.º 2 | Mayo-Agosto 2026 | PP 5-15

DOI: 10.20983/culcyt.2026.2.2.1

ISSN (electrónico) 2007-0411

e23201

ARTÍCULO DE INVESTIGACIÓN

Aplicación de programación entera binaria para el balanceo de un proceso productivo automotriz

Application of Binary Integer Programming for the Balancing of an Automotive Production Process

Antonio Alejandro Morales Velasco[1], *Luis Alberto Rodríguez-Picón*[1] ✉, *Karla Gabriela Gómez Bull*[1]
[1] Departamento de Ingeniería Industrial y Manufactura, Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez; Ciudad Juárez, Chihuahua, México

RESUMEN

El balanceo de líneas permite distribuir actividades entre estaciones de trabajo considerando restricciones de tiempo y precedencia. Este estudio analiza el balanceo de un proceso de ensamble automotriz mediante la aplicación secuencial de métodos heurísticos y programación entera binaria. Se utilizó un caso de referencia compuesto por 33 actividades, con un contenido total de trabajo de 89.28 min y un *takt time* de 18.018 min. Inicialmente, el método del candidato más largo produjo una solución factible de seis estaciones, resultado que se utilizó para limitar el número de estaciones consideradas en la formulación binaria y reducir las variables de decisión de 1089 a 198. El modelo fue implementado en R y obtuvo igualmente una solución de seis estaciones con la misma asignación de actividades. Como análisis complementario, el método de Helgeson-Birnie también produce seis estaciones, aunque con diferencias parciales en la distribución de las actividades. Los resultados muestran que distintas estrategias de balanceo pueden converger al mismo número de estaciones y que una solución heurística inicial puede utilizarse para reducir el tamaño de la formulación binaria. Desde una perspectiva práctica, este enfoque puede apoyar la toma de decisiones en procesos de ensamble automotriz al facilitar la evaluación de configuraciones de estaciones de trabajo y la identificación de oportunidades para una utilización más equilibrada de los recursos productivos.

PALABRAS CLAVE: balanceo de línea; método heurístico; programación entera binaria; *takt time*; tiempo de ciclo.

ABSTRACT

Line balancing allows activities to be distributed among workstations while considering time and precedence constraints. This study analyzes the balancing of an automotive assembly process through the sequential application of heuristic methods and binary integer programming. A reference case consisting of 33 activities was used, with a total work content of 89.28 min and a takt time of 18.018 min. Initially, the longest candidate rule produced a feasible six-workstation solution, which was then used to limit the number of workstations considered in the binary formulation and reduce the number of decision variables from 1,089 to 198. The model was implemented in R and also produced a six-workstation solution with the same task assignment. As a complementary analysis, the Helgeson-Birnie method also resulted in six workstations, although with partial differences in task allocation. The results show that different balancing strategies may converge to the same number of workstations and that an initial heuristic solution can be used to reduce the size of the binary formulation. From a practical perspective, this approach can support decision-making in automotive assembly processes by facilitating the evaluation of workstation configurations and the identification of opportunities for a more balanced use of production resources.

KEYWORDS: line balancing; heuristic method; binary integer programming; takt time; cycle time.

Correspondencia:
DESTINATARIO: Luis Alberto Rodríguez-Picón
INSTITUCIÓN: Universidad Autónoma de Ciudad Juárez / Instituto de Ingeniería y Tecnología
DIRECCIÓN: Av. del Charro núm. 450 norte, C. P. 32320, Ciudad Juárez, Chihuahua, México
CORREO ELECTRÓNICO: luis.picon@uacj.mx

Fecha de recepción: 14 de abril de 2026. Fecha de aceptación: 20 de julio de 2026. Fecha de publicación: 21 de agosto de 2026.

acceso abierto UACJ

REVISTA ARBITRADA

I. INTRODUCCIÓN

En la actualidad, la competitividad industrial exige que sus procesos de producción sean óptimos y balanceados para responder a la alta demanda del mercado. Sin embargo, una de las problemáticas actuales radica en que las líneas de producción y manufactura son ineficientes debido a la falta de equilibrio de tiempos, cuellos de botella, acumulación de inventario y procesos ineficientes, entre otros [1].

En este contexto, la industria automotriz constituye un sector particularmente relevante debido a sus altos volúmenes de producción y a la necesidad de mantener procesos de ensamble coordinados y eficientes. En México, durante 2024 se produjeron 3 989 403 vehículos ligeros, lo que representó un incremento de 5.56 % respecto al año anterior [2]. Este nivel de producción evidencia la importancia de disponer de herramientas que permitan analizar la asignación de actividades y el aprovechamiento de las estaciones de trabajo en procesos de manufactura caracterizados por altos volúmenes de operación.

Existen diversos estudios que destacan herramientas como Value Stream Mapping (VSM), utilizada para comprender el flujo de materiales a través de la cadena de suministro. Por otra parte, existen metodologías como Kaizen, que consisten en la mejora continua, mediante la cual se busca el mejor desempeño en el proceso. Asimismo, existen recursos como el SMED (Single Minute Exchange of Die) que busca reducir los tiempos de preparación y cambios de modelos, minimizando la carga y mejorando la sincronización del flujo productivo [3].

En este sentido, otro método importante para mejorar los indicadores de procesos productivos, así como la competitividad, consiste en el balanceo de líneas de ensamble. El balanceo de líneas es una técnica de ingeniería industrial cuyo propósito es contribuir en tareas de distintas estaciones de trabajo de las líneas de producción. Su principal objetivo es lograr que los tiempos se acerquen lo más posible al *takt time* (TT), evitar la presencia de tiempos muertos, así como la inactividad de equipos, por lo que el balanceo de líneas favorece un flujo de producción continuo, eficiente y en sincronía [4].

Una característica de este método es que depende en gran medida de las magnitudes y variaciones de los tiempos de ciclo para un balanceo apropiado, dado que se busca que los tiempos totales de tareas asignadas a cada estación no excedan el tiempo del TT [5]. Un elemento esencial en el balanceo del proceso es la flexibilidad, dado que este proceso permite adaptarse a cambios, tanto de distribución de equipos como de demanda de producción [6]. En ocasiones se pueden presentar volúmenes altos de demanda; por lo tanto, se requiere que el proceso obtenga un equilibrio adecuado, procurando reducir los tiempos muertos al mínimo. Otro factor a tomar en cuenta es la asignación de diferentes tareas que se deben realizar respetando la precedencia lógica de cada estación hasta la última actividad del proceso [7].

Entre los beneficios importantes del método de balanceo de líneas está la reducción de cuellos de botella, lo que mejora significativamente la producción [8]. Se ha comprobado que la aplicación de herramientas de este tipo impulsa la productividad y mejora la capacidad de respuesta ante la demanda, al mismo tiempo que favorece un mejor control de costos [9]. Este método permite aumentar la producción sin necesidad de incrementar la mano de obra, además de facilitar una inversión más precisa en equipos de producción, basada en la optimización de recursos [10]. Por consiguiente, se permite un flujo continuo, lo cual a su vez reduce los inventarios en los procesos [11].

En este sentido, resulta relevante discutir métodos específicos de balanceo de líneas. Un método relevante es la programación lineal mixta, que consiste en reducir la cantidad de estaciones de ciclo mediante diversos algoritmos y número de estaciones con modelos matemáticos [12]. Este método genera soluciones óptimas que incorporan algunas restricciones, como costos, tiempos máximos y precedencias, entre otros [13]. Asimismo, otro método utilizado es la programación entera binaria, cuyo funcionamiento es que las líneas de ensamble se analicen de manera secuencial utilizando 1 y 0. Una ventaja es que la programación binaria permite la realización de variables con programación lineal con un enfoque computacional [14]. A continuación, se presentan algunos otros métodos:

Programación por restricciones. Este es un método que considera las precedencias. Si una estación termina y requiere una precedencia, entonces se hace un balanceo en donde se toman en cuenta tiempos lógicos y matemáticas simples. Con esta alternativa se busca obtener soluciones satisfactorias con todas las restricciones [15].

Enumeración exhaustiva. Con este método se evalúan todas las posibles combinaciones de estaciones de trabajo hasta encontrar la solución óptima, lo que garantiza un mejor balanceo [16].

Ramificación y acotación. En este método, del inglés Branch and Bound, se explora sistemáticamente el espacio de soluciones mediante la división del problema original en subproblemas o ramas. Para cada subproblema se calculan cotas que permiten determinar si la región correspondiente puede contener una solución mejor que la mejor solución factible encontrada hasta ese momento. Cuando una rama no puede mejorar dicha solución, se descarta, reduciendo así la cantidad de combinaciones que deben evaluarse [17].

Métodos de algoritmo heurístico. Están basados en el rango de compresión y un ejemplo de aplicación ha sido presentado por [18]. En este estudio heurístico, los autores proponen balancear líneas de ensamble en forma de U, que se basa en calcular el valor del rango integral, lo que permite a su vez asignar las estaciones respetando las precedencias y tiempos de ciclos con métodos de algoritmos que son significativamente más exactos en los cálculos. Por tanto, en este método se proponen estrategias para balancear líneas de ensambles bidireccionales y dinámicas, lo que permite minimizar el número de estaciones de trabajo para evitar la tasa de tiempo de inactividad de cada estación [19].

Dadas estas consideraciones, el objetivo de este trabajo es analizar el balanceo de una línea de producción a partir de sus tiempos de ciclo y relaciones de precedencia, mediante la aplicación de métodos heurísticos y programación entera binaria. Como caso de referencia, se consideran los datos del proceso de ensamble automotriz reportados por Saurabh y Salman [20]. En particular, del estudio original se toman las 33 actividades del proceso, sus tiempos de ciclo y las relaciones de precedencia. Estos datos se utilizan como caso de referencia para desarrollar y evaluar el esquema de balanceo propuesto.

A partir de esta información se propone una solución al balanceo del proceso considerando el método heurístico de candidato más largo y, después, la solución encontrada se toma en cuenta para reducir el número de variables de decisión en un esquema de optimización basado en programación entera binaria. De esta manera, la aportación del estudio se centra en utilizar la solución heurística inicial para acotar la formulación de programación entera binaria y posteriormente analizar la correspondencia entre ambas soluciones.

II. METODOLOGÍA

En esta sección se presenta la metodología utilizada para analizar el balanceo del proceso. El procedimiento se desarrolla en tres etapas principales: caracterización del caso de referencia, obtención de una solución inicial mediante el método heurístico del candidato más largo y formulación del problema mediante programación entera binaria. Finalmente, se comparan los resultados obtenidos con ambos enfoques. La secuencia general del procedimiento se muestra en la Figura 1.

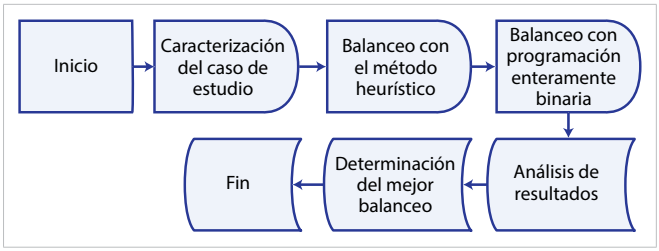

Figura 1. Diagrama del método propuesto.

A continuación, se describen los puntos que se desarrollan en esta metodología

1. Caracterización del caso de estudio. En este artículo se utilizaron los datos del proceso de ensamble automotriz publicados por Saurabh y Salman [20]. Del estudio original se tomaron las 33 actividades, los tiempos de ciclo y sus relaciones de precedencia. Por tanto, esta información abarca los datos de entrada para la aplicación de la estrategia de solución desarrollada en el presente trabajo. A partir de las relaciones de precedencia reportadas se estableció el diagrama correspondiente que fue utilizado tanto por el método heurístico como por el modelo de programación entera binaria.

2. Balanceo con método heurístico. Una vez establecido el diagrama de precedencias, se realizó el balanceo mediante el método del candidato más largo. Este procedimiento ordena las actividades de mayor a menor tiempo de operación y las asigna secuencialmente a las estaciones de trabajo, respetando las relaciones de precedencia y procurando que la carga total de cada estación no exceda el TT definido en (1). Cuando una actividad no puede ser asignada a la estación actual debido al límite de tiempo o a una

restricción de precedencia, se evalúa la siguiente actividad factible.

$$\text{TT} = \frac{\textit{tiempo disponible}}{\textit{demanda}} \quad (1)$$

En este método se ordenan las tareas con mayor a menor tiempo más largo al más pequeño Se tiene que respetar las precedencias; hay tareas que tienen ciertas restricciones en la precedencia y otras no pueden iniciar hasta que las relaciones de precedencias se completen en dichas tareas.

El método heurístico se utiliza inicialmente para obtener una solución factible y establecer un límite superior para el número de estaciones del problema. Sea m_H el número de estaciones obtenido mediante el método del candidato más largo y m^* el número mínimo de estaciones. Debido a que la solución heurística satisface las restricciones de tiempo de ciclo y precedencia, se cumple $m^* \leq m_H$. Por otra parte, un límite inferior para el número de estaciones puede establecerse a partir del contenido total de trabajo como,

$$m_L = \left\lceil \frac{\sum_{i=1}^{n} t_i}{\text{TT}} \right\rceil$$

De esta manera, el número óptimo de estaciones se encuentra acotado por,

$$m_L \leq m^* \leq m_H$$

El límite superior m_H se utiliza para definir el número máximo de estaciones candidatas en el modelo de programación entera binaria. De esta manera, para n actividades, la formulación requiere como máximo $n \times m_H$ variables binarias, en lugar de considerar variables bajo el supuesto conservador de hasta una estación potencial por actividad. Por lo tanto, el método heurístico proporciona información para acotar su formulación.

3. Balanceo con programación entera binaria. En este paso se pretende balancear el proceso considerando programación entera binaria. Se consideró definir una función objetivo como en (2),

$$\text{mín: } Z = X_1 + 2X_2 + 3X_3 + \cdots + kX_m \quad (2)$$

donde k indica el número de estaciones obtenidas en el paso 2 de la metodología propuesta, dado que esto permite reducir el número de variables de decisión. El modelo considera cuatro tipos de restricciones: asignación, TT, precedencia y binariedad. A continuación, se presenta cada conjunto de restricciones. La primera de ellas, que es sobre la asignación de operaciones e implica que cada operación sea asignada solo una vez a una estación, tienen la forma que se presenta en (3):

$$X_1 + X_2 + \cdots X_k = 1 \quad (3)$$

Las siguientes restricciones son las de TT presentadas en (4), en las cuales se considerará que la suma de las operaciones asignadas a una estación no exceda el TT del proceso:

$$TC_1X_1 + TC_2X_2 + \cdots TC_kX_k \leq \text{TT} \quad (4)$$

donde TC_1 implica el tiempo de ciclo de la operación X_1. El tercer conjunto de restricciones tiene que ver con las precedencias, en las cuales se asegura que una operación no sea asignada a una estación si sus precedencias no han sido asignadas anteriormente y tiene la siguiente forma:

$$X_1 \leq X_{1+k}$$
$$X_2 \leq X_{1+k} + X_{2+k}$$
$$X_3 \leq X_{1+k} + X_{2+k} + \cdots + X_{m+k}$$

Por último, la restricción que asegura que todas las variables de decisión sean variables binarias es X_1, X_2, .. $X_{m,k}$ = binarias(0,1), lo que implica que $X_i = 1$ cuando una actividad es asignada a una estación y $X_i = 0$ cuando la actividad no es asignada a la estación.

4. Análisis de resultados. El modelo de programación lineal entera binaria formulado en la etapa anterior fue implementado y resuelto mediante el lenguaje de programación R, utilizando RStudio Desktop como entorno de desarrollo integrado (IDE) y el paquete lpSolve para la solución del modelo. Se seleccionó lpSolve debido a que permite formular y resolver problemas de programación lineal entera y binaria dentro del entorno de R y sus capacidades resultan suficientes para el tamaño del problema analizado. En este caso, se tienen 198 variables binarias de decisión, además de las restricciones de asignación, *takt time* y precedencia.

Si bien solucionadores como Gurobi o CPLEX ofrecen capacidades computacionales avanzadas para problemas de mayor escala, dichas características no

son necesarias para la formulación en este estudio. Una vez obtenida la solución del modelo, se analizaron los valores de las variables de decisión y de la función objetivo, con el propósito de evaluar la solución obtenida.

5. En esta etapa se evaluaron los resultados obtenidos mediante el método del candidato más largo y el modelo de programación entera binaria. Para ello, se compararon las soluciones considerando el número de estaciones de trabajo requeridas, la distribución de las tareas entre las estaciones y los tiempos de ocio resultantes. Esta comparación permitió determinar si la solución heurística alcanzó el número mínimo de estaciones establecido por el modelo exacto, así como identificar posibles mejoras en el balanceo de la línea. Asimismo, se analizó el efecto de utilizar la solución obtenida mediante el método del candidato más largo como límite superior para reducir el tamaño de la formulación del modelo de programación entera binaria y facilitar su proceso de solución.

III. RESULTADOS Y DISCUSIÓN

El presente caso de referencia se basa en un proceso de la industria de automotriz, en donde se fabrican componentes de chapa metálica, con ensambles metálicos, como soldaduras y ensambles, y a la vez se monta en el chasis de la carrocería. Para evaluar la estrategia propuesta se utilizó el caso de referencia descrito anteriormente. La Figura 2 y la Tabla 1 presentan, respectivamente, las relaciones de precedencia y los tiempos de ciclo reportados para las 33 actividades del proceso. A partir de estos datos se desarrollaron en este trabajo dos etapas de solución: primera, el balanceo mediante el método heurístico del candidato más largo y, posteriormente, la formulación y solución del modelo de programación entera binaria.

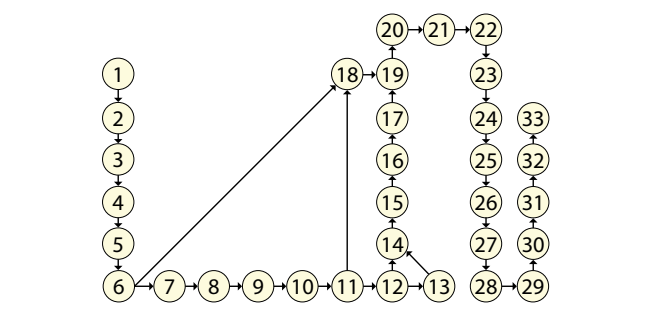

Figura 2. Diagrama de precedencias del caso de referencia, elaborado a partir de [20].

En la Tabla 1 se muestran los tiempos de ciclo de cada operación del proceso considerado, en la cual se puede ver la información relacionada a los elementos y precedencias de cada actividad.

TABLA 1
ACTIVIDADES, TIEMPOS DE CICLO Y RELACIONES DE PRECEDENCIA [20]

ACTIVIDAD	ELEMENTOS	TIEMPO DE CICLO (min)	PRECEDENCIA
Línea 1	1	2.46	---
Línea 2	2	2.58	1
Línea 3	3	3.15	2
Línea 4	4	2.37	3
Línea 5	5	3.2	4
Línea 6	6	2.03	5
Panel frontal	7	1.37	---
Geo de panel frontal	8	3.08	7
Mecanizado permanente puntual 1	9	2.5	8
Mecanizado permanente puntual 2	10	1.11	9
Accesorio de inversión 1	11	0.11	10
Panel lateral izquierdo	12	2.51	---
Panel lateral derecho	13	2.24	---
Mecanizado permanente puntual 1	14	1.15	12, 13
Mecanizado permanente puntual 2	15	1.1	14
Punto de fijado permanente	16	2.36	15
Accesorio de inversión 2	17	3.26	16
Línea principal 1	18	2.39	6,11
Línea principal 2	19	4.13	17, 18
Línea principal 3	20	3.18	19
Línea principal 4	21	4	20
Línea principal 5	22	3.47	21
Línea principal 6	23	3.32	22
Línea principal 7	24	3.1	23
Línea principal 8	25	2.37	24
Línea principal 9	26	2.33	25
Cubierta alta 1	27	2.39	26
Cubierta alta 2	28	2.3	27
Cubierta alta 3	29	2.38	28
Cubierta alta 4	30	2.1	29
Cubierta alta 5	31	3.24	30
Cubierta alta 6	32	2.53	31
Cubierta alta 7	33	9.47	32

En primera instancia se realizó un balanceo mediante el método heurístico de candidato más largo, para este

caso se consideró un TT de 18.018. Como resultado se obtuvo que el proceso se puede dividir en 6 respectivas estaciones, cada una respetando las precedencias del candidato más largo y sin exceder el TT, en total con una suma de 89.28 minutos en total. Los resultados se muestran en la Tabla 2. Mientras que en la Figura 3 se presenta una ilustración del balanceo de las actividades por cada estación de trabajo.

TABLA 2
BALANCEO DEL PROCESO CON EL MÉTODO DEL CANDIDATO MÁS LARGO

ESTACIÓN	ACTIVIDAD	SUMA DE TIEMPOS	TC
1	1, 2, 3, 4, 5, 6, 7	2.46 + 2.58 + 3.15 + 2.37 + 3.2 + 2.03 + 2.39	17.16
2	8, 9, 10, 11, 12, 18, 13, 14, 15	3.08 + 2.5 + 1.11 + 0.11 + 2.51 + 2.39 + 2.24 + 1.15 + 1.1	16.19
3	16, 17, 19, 20, 21	2.36 + 3.26 + 4.13 + 3.18 + 4	16.93
4	22, 23, 24, 25, 26, 27	3.47 + 3.32 + 3.1 + 2.37 + 2.33 + 2.39	16.98
5	28, 29, 30, 31, 32	2.3 + 2.38 + 2.1 + 3.24 + 2.53	12.55
6	33	9.47	9.47
Suma de tiempos			**89.28**

El balanceo obtenido a través método del candidato más largo resultó como se puede ver en la Figura 3.

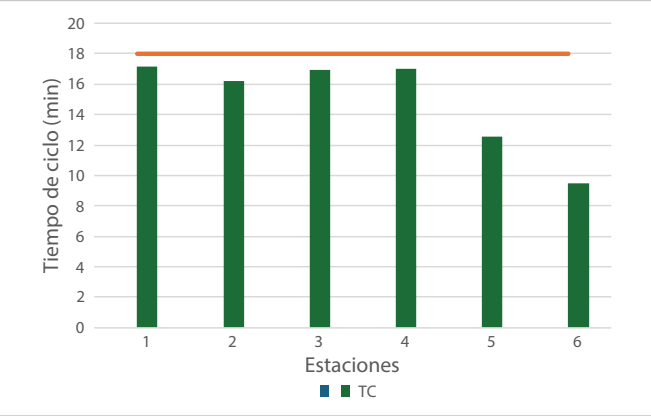

Figura 3. Tiempos de ciclo por estación.

El método del candidato más largo produjo una solución factible con seis estaciones, por lo que $m_H = 6$. Por otra parte, considerando el tiempo total de trabajo de 89.28 min y un TT de 18.018 min, el límite inferior resulta como $m_L = [89.28/18.018] = 5$.

Por ende, antes de aplicar el método exacto puede establecerse que $m_L \leq m^* \leq m_H$. La solución heurística permitió limitar la formulación exacta a un máximo de seis estaciones candidatas. Al considerar las 33 actividades del proceso, esto reduce el número de variables binarias potenciales de 1089 a 198, equivalente a una reducción de 81.82 %.

Una vez tomando esto en cuenta, se procedió a considerar el método exacto de programación entera binaria, bajo las siguientes consideraciones: se tienen 33 (i) actividades, como se puede notar en la Tabla 1, e inicialmente se considera una estación por actividad lo cual implica 33 (j) estaciones para un total de $33 \times 33 = 1089$ variables de decisión. Para este caso de estudio, dado que con el método del candidato más largo se obtuvieron 6 (j) estaciones, entonces se redujo el número de variables de decisión a $33 \times 6 = 198$. Por tanto, las variables de decisión están definidas como sigue:

$$X_{i,j}: \left\{ \begin{array}{c} X_{1,1}, X_{1,2}, X_{1,3}, X_{1,4}, X_{1,5}, X_{1,6}, \\ X_{2,1}, X_{2,2}, X_{2,3}, X_{2,4}, X_{2,5}, X_{2,6}, \\ \vdots \\ X_{33,1}, X_{33,2}, X_{33,3}, X_{33,4}, X_{33,5}, X_{33,6} \end{array} \right\} \quad (5)$$

Dado que $X_{i,j}$ = binarias(0,1), cuando $X_{i,j} = 1$ implica que la actividad i es asignada a la estación j, y cuando $X_{i,j} = 0$ la actividad i no es asignada. En este sentido la función objetivo del problema de programación está definida de la siguiente manera:

$$\begin{aligned} \text{mín: } Z = {} & X_{1,1} + 2X_{1,2} + 3X_{1,3} + 4X_{1,4} + 5X_{1,5} + 6X_{1,6} + \\ & X_{2,1} + 2X_{2,2} + 3X_{2,3} + 4X_{2,4} + 5X_{2,5} + 6X_{2,6} + \cdots + \\ & X_{33,1}, 2X_{33,2}, 3X_{33,3}, 4X_{33,4}, 5X_{33,5}, 6X_{33,6} \end{aligned} \quad (6)$$

con lo cual se buscó minimizar Z, que representa el número de estaciones para el proceso. De igual manera se consideraron pesos posicionales 1, 2, …, 6 para cada grupo $X_{i,1}, X_{i,2}, \ldots, X_{i,6}$; $i = 1, 2, \ldots, 33$ con la intención de procurar que cada operación sea asignada inmediatamente a la primera estación disponible y evitar, por ejemplo, que la primera operación sea asignada hasta una estación muy posterior.

Con base en el paso 4 del método propuesto, a continuación se desarrollaron las restricciones de asignación, las cuales resultaron de la siguiente manera:

$$X_{1,1} + X_{1,2} + X_{1,3} + X_{1,4} + X_{1,5} + X_{1,6} = 1$$
$$X_{2,1} + X_{2,2} + X_{2,3} + X_{2,4} + X_{2,5} + X_{2,6} = 1$$
$$X_{3,1} + X_{3,2} + X_{3,3} + X_{3,4} + X_{3,5} + X_{3,6} = 1$$
$$\vdots$$
$$X_{33,1} + X_{33,2} + X_{33,3} + X_{33,4} + X_{33,5} + X_{33,6} = 1$$

Con estas se aseguró que cada operación fuera asignada solo una vez a cada estación para obtener un total de 33 restricciones de este tipo. A continuación, se muestran las restricciones relacionadas con el TT, en las cuales se tomó en cuenta los tiempos de ciclo de cada actividad de la Tabla 1 y se buscó que la suma de los tiempos de ciclo de cada actividad que es asignada a las estaciones no excediera el tiempo de 18.018 minutos. Estas restricciones resultaron de la siguiente manera:

$$2.46X_{1,1} + 2.58X_{2,1} + 3.15X_{3,1} + \ldots + 9.47X_{33,1} \leq 18.018$$
$$2.46X_{1,2} + 2.58X_{2,2} + 3.15X_{3,2} + \ldots + 9.47X_{33,2} \leq 18.018$$
$$2.46X_{1,3} + 2.58X_{2,3} + 3.15X_{3,3} + \ldots + 9.47X_{33,3} \leq 18.018$$
$$2.46X_{1,4} + 2.58X_{2,4} + 3.15X_{3,4} + \ldots + 9.47X_{33,4} \leq 18.018$$
$$2.46X_{1,5} + 2.58X_{2,5} + 3.15X_{3,5} + \ldots + 9.47X_{33,5} \leq 18.018$$
$$2.46X_{1,6} + 2.58X_{2,6} + 3.15X_{3,6} + \ldots + 9.47X_{33,6} \leq 18.018$$

Por último, las restricciones de precedencias, en el diagrama de la Figura 2 se tienen un total de 35 relaciones de precedencia y para cada una de ellas fue necesario definir un conjunto de restricciones de manera que se asegure que una actividad inicial se asignada primeramente antes de la actividad después de ella.

En este sentido, a continuación se presentan los conjuntos de restricciones para las primeras 5 relaciones de precedencia.

Restricciones para la precedencia entre la actividad 1 y 2:

$$X_{1,1} \leq X_{2,1} + X_{2,2} + X_{2,3} + X_{2,4} + X_{2,5} + X_{2,6}$$
$$X_{1,2} \leq X_{2,2} + X_{2,3} + X_{2,4} + X_{2,5} + X_{2,6}$$
$$X_{1,3} \leq X_{2,3} + X_{2,4} + X_{2,5} + X_{2,6}$$
$$X_{1,4} \leq X_{2,4} + X_{2,5} + X_{2,6}$$
$$X_{1,5} \leq X_{2,5} + X_{2,6}$$
$$X_{1,6} \leq X_{2,6}$$

Restricciones para la precedencia entre la actividad 2 y 3:

$$X_{2,1} \leq X_{3,1} + X_{3,2} + X_{3,3} + X_{3,4} + X_{3,5} + X_{3,6}$$
$$X_{2,2} \leq X_{3,2} + X_{3,3} + X_{3,4} + X_{3,5} + X_{3,6}$$
$$X_{2,3} \leq X_{3,3} + X_{3,4} + X_{3,5} + X_{3,6}$$
$$X_{2,4} \leq X_{3,4} + X_{3,5} + X_{3,6}$$
$$X_{2,5} \leq X_{3,5} + X_{3,6}$$
$$X_{2,6} \leq X_{3,6}$$

Restricciones para la precedencia entre la actividad 3 y 4:

$$X_{3,1} \leq X_{4,1} + X_{4,2} + X_{4,3} + X_{4,4} + X_{4,5} + X_{4,6}$$
$$X_{3,2} \leq X_{4,2} + X_{4,3} + X_{4,4} + X_{4,5} + X_{4,6}$$
$$X_{3,3} \leq X_{4,3} + X_{4,4} + X_{4,5} + X_{4,6}$$
$$X_{3,4} \leq X_{4,4} + X_{4,5} + X_{4,6}$$
$$X_{3,5} \leq X_{4,5} + X_{4,6}$$
$$X_{3,6} \leq X_{4,6}$$

Restricciones para la precedencia entre la actividad 4 y 5:

$$X_{4,1} \leq X_{5,1} + X_{5,2} + X_{5,3} + X_{5,4} + X_{5,5} + X_{5,6}$$
$$X_{4,2} \leq X_{5,2} + X_{5,3} + X_{5,4} + X_{5,5} + X_{5,6}$$
$$X_{4,3} \leq X_{5,3} + X_{5,4} + X_{5,5} + X_{5,6}$$
$$X_{4,4} \leq X_{5,4} + X_{5,5} + X_{5,6}$$
$$X_{4,5} \leq X_{5,5} + X_{5,6}$$
$$X_{4,6} \leq X_{5,6}$$

Restricciones para la precedencia entre la actividad 5 y 6:

$$X_{5,1} \leq X_{6,1} + X_{6,2} + X_{6,3} + X_{6,4} + X_{6,5} + X_{6,6}$$
$$X_{5,2} \leq X_{6,2} + X_{6,3} + X_{6,4} + X_{6,5} + X_{6,6}$$
$$X_{5,3} \leq X_{6,3} + X_{6,4} + X_{6,5} + X_{6,6}$$
$$X_{5,4} \leq X_{6,4} + X_{6,5} + X_{6,6}$$
$$X_{5,5} \leq X_{6,5} + X_{6,6}$$
$$X_{5,6} \leq X_{6,6}$$

De esta manera se estableció las restricciones para el resto de las 30 relaciones de precedencia que se puede observar tanto en la Figura 2 como en la Tabla 1.

Si se considera que por cada relación de precedencia se tienen 6 restricciones, entonces se dispone de un total de $35 \times 6 = 210$ restricciones solo para las precedencias.

El problema de optimización descrito anteriormente se solucionó en el software R versión 4.5.1, considerando el paquete "lpSolve". El código consiste en diferentes componentes que a continuación se describen.

En la parte 1 se declaran los coeficientes de la función objetivo de acuerdo con (6). De esta manera, la función resultó como sigue:

```
f.obj <- c(1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,.
..)
```

En la siguiente parte del código se consideraron los coeficientes de todas las restricciones del problema de optimización, tomando en cuenta las 6 restricciones de TT, las 210 de precedencias y las 33 de asignación. En general se conformó un arreglo de coeficientes en la siguiente función:

```
f.con <- matrix(c("Coeficientes de las
restricciones", nrow = 249, byrow = TRUE)
```

En las siguientes líneas de código se plantean las desigualdades de todas las restricciones, así como el lado derecho de las mismas, de la siguiente manera:

```
f.dir <- c("<=","<=",...)
f.rhs <- c(18.018, 18.018,...)
```

Al final, el problema es resulto a través de la siguiente función:

```
lp("min", f.obj, f.con, f.dir, f.rhs, int.vec
= 1:198, all.bin = TRUE, num.bin.solns=1)
```

Al final, el problema fue resuelto a través de la siguiente función:

```
lp("min", f.obj, f.con, f.dir, f.rhs, int.vec
= 1:198, all.bin = TRUE, num.bin.solns=1)
```

Note que se considera la función `lp`, la cual forma parte del paquete "lpSolve". Se indica el objetivo de minimizar, dado que se desea obtener el menor número posible de estaciones.

También se consideraron los elementos del problema de optimización descritos anteriormente como `f.obj`, `f.con`, `f.dir`, `f.rhs`.

Se declaró que todas las variables de decisión son enteras y binarias en `int.vec = 1:198` y `all.bin = TRUE`, respectivamente. Al correr el código se obtuvieron los siguientes resultados:

TABLA 3
SOLUCIÓN OBTENIDA MEDIANTE PROGRAMACIÓN ENTERA BINARIA EN R

1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6	1 2 3 4 5 6
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0	0 1 0 0 0 0
0 1 0 0 0 0	0 1 0 0 0 0	0 1 0 0 0 0	0 1 0 0 0 0
0 1 0 0 0 0	0 1 0 0 0 0	0 1 0 0 0 0	0 0 1 0 0 0
0 0 1 0 0 0	0 1 0 0 0 0	0 0 1 0 0 0	0 0 1 0 0 0
0 0 1 0 0 0	0 0 0 1 0 0	0 0 0 1 0 0	0 0 0 1 0 0
0 0 0 1 0 0	0 0 0 1 0 0	0 0 0 1 0 0	0 0 0 0 1 0
0 0 0 0 1 0	0 0 0 0 1 0	0 0 0 0 1 0	0 0 0 0 1 0
0 0 0 0 0 1			

Como se puede observar en la Tabla 3, el esquema de optimización arrojó que los resultados obtenidos son iguales al candidato más largo en la Tabla 2. Específicamente, el modelo de programación entera binaria obtuvo como solución óptima seis estaciones, por lo que $m^* = 6$. Este resultado permitió cerrar el intervalo establecido previamente $5 \leq m^* \leq 6$, y confirmó que la solución de seis estaciones obtenida mediante el candidato más largo alcanza el número mínimo de estaciones para las condiciones del caso analizado.

En consecuencia, la utilidad de la estrategia secuencial no radica en que el modelo exacto genere necesariamente una asignación diferente de la heurística, sino en utilizar una solución heurística factible para acotar la formulación y posteriormente verificar la optimalidad mediante programación entera binaria. En la Tabla 4 se presenta un resumen de las actividades y sus respectivos tiempos de ciclo. Al final de cada estación se muestra la suma de los tiempos de las actividades, por lo que se puede notar que ninguna excede el TT establecido.

TABLA 4
ASIGNACIÓN DE ACTIVIDADES POR ESTACIONES

Estación											
1		2		3		4		5		6	
1	2.46	**8**	3.08	**16**	2.36	**22**	3.47	**28**	2.3	**33**	9.47
2	2.58	**9**	2.5	**17**	3.26	**23**	3.32	**29**	2.38		
3	3.15	**10**	1.11	**19**	4.13	**24**	3.1	**30**	2.1		
4	2.37	**11**	0.11	**20**	3.18	**25**	2.37	**31**	3.24		
5	3.2	**12**	2.51	**21**	4	**26**	2.33	**32**	2.53		
6	2.03	**13**	2.24			**27**	2.39				
7	1.37	**14**	1.15								
		15	1.1								
		18	2.39								
17.16		**16.19**		**16.93**		**16.98**		**12.55**		**9.47**	

Los resultados de balanceo obtenidos deben interpretarse en función de las características del problema bajo estudio y no como evidencia de superioridad de alguno de los métodos. En el caso analizado, tanto el candidato más largo como Helgeson-Birnie y la programación entera binaria conducen a una solución de seis estaciones, aunque las dos estrategias heurísticas no producen exactamente la misma asignación de actividades (solución a través de Helgeson-Birnie reportada por [20]). El método de Helgeson-Birnie incorpora explícitamente la estructura de precedencias mediante el cálculo de pesos posicionales, por lo que puede generar prioridades de asignación diferentes a las del candidato más largo [21].

Por otra parte, la literatura reciente muestra que los problemas de balanceo de líneas han evolucionado hacia modelos con restricciones adicionales, múltiples objetivos y configuraciones productivas más complejas, lo que ha motivado el desarrollo de métodos híbridos, exactos y metaheurísticos [22]. En particular, los algoritmos genéticos han mostrado utilidad para explorar espacios de solución de gran tamaño y manejar restricciones industriales específicas, incluyendo aplicaciones recientes en líneas automotrices [23]. De manera similar, se han desarrollado algoritmos genéticos mejorados para problemas de balanceo con restricciones de recursos y configuraciones de dos lados [24].

En contraste, los modelos de programación entera permiten representar explícitamente las restricciones de asignación, tiempo de ciclo y precedencia, aunque el tamaño de la formulación puede incrementarse conforme crece el número de actividades y estaciones consideradas. En este sentido, los resultados del presente caso muestran que diferentes estrategias pueden converger al mismo número de estaciones, pero con estructuras de asignación distintas.

IV. CONCLUSIONES

En este artículo se llevó a cabo un balanceo de línea de ensamble sobre un caso de referencia correspondiente a un proceso de ensamble automotriz. Se consideró inicialmente un método heurístico para después llevar a cabo un esquema de optimización basado en programación entera binaria.

La aplicación inicial del método del candidato más largo permitió obtener una solución factible de seis estaciones y, con ello, establecer un límite superior para la formulación exacta. En conjunto con el límite inferior de cinco estaciones obtenido a partir del tiempo total de trabajo y el TT, se estableció que la solución óptima debía encontrarse entre cinco y seis estaciones. El uso del límite superior heurístico permitió reducir la formulación de 1089 a 198 variables binarias, equivalente a una reducción del 81.82 %. Posteriormente, el modelo de programación entera binaria determinó un mínimo de seis estaciones, confirmando que la solución heurística alcanzaba el número mínimo de estaciones para las condiciones y restricciones del caso analizado

El problema fue resuelto mediante programación entera binaria utilizando el software R, obteniéndose una solución de seis estaciones. Este resultado coincide con la solución obtenida previamente mediante el método heurístico del candidato más largo, lo que muestra que, para las condiciones del caso analizado, la solución heurística y la obtenida mediante el modelo exacto presentan el mismo número de estaciones y la misma asignación de actividades.

Por supuesto, las condiciones de balanceo dependen de múltiples factores, por ejemplo, la variación de los tiempos de ciclo de las estaciones (algunas con tiempos muy pequeños y otras con tiempos de ciclo muy grandes) y las relaciones de precedencia. En este caso se notó que la última actividad del proceso, llamada Cubierta ala 7, cubre más del 50 % del TT. Si resultara necesario mejorar aún más el balanceo del proceso, sería necesario reducir el tiempo de ciclo de esta última actividad. Para esto se pueden considerar diversas herramientas de manufactura esbelta, con la intención de reducir directamente el tiempo a través de la reducción de desperdicios y a través de la subdivisión de la actividad en múltiples operaciones con menores tiempos de ciclo.

Como trabajo futuro, se puede explorar la estrategia analizada en líneas de producción de mayor tamaño y con condiciones operativas más dinámicas, incluyendo variaciones en la demanda, tiempos de operación variables, líneas flexibles y configuraciones con múltiples productos. También sería pertinente comparar experimentalmente el desempeño de la formulación binaria con otros métodos heurísticos y metaheurísticos en términos de calidad de la solución, tiempo computacional y escalabilidad.

REFERENCIAS

[1] E. I. Borja-Salinas, M. Ortuño-Delgado y L. Giraldo-Ceballos, "Prioridades competitivas en la producción: configuración en las industrias del Ecuador", *Sinergia Académica*, vol. 7, n.º 4, pp. 621-633, oct. 2024, doi: 10.51736/8qt7yd90.

[2] Instituto Nacional de Estadística y Geografía, "Registro Administrativo de la Industria Automotriz de Vehículos Ligeros (RAIAVL)", INEGI. [En línea]. Disponible: https://www.inegi.org.mx/datosprimarios/iavl/

[3] J. A. Valdebenito, "Implementación de Lean Manufacturing y Metodologías de Mejora Continua: Adaptación de Estándares Globales para la Industria Nacional", tesis de maestría, Universidad del Desarrollo, 2025. [En línea]. Disponible: https://repositorio.udd.cl/items/2c569ff9-acad-4b21-953a-d12f59759ea2

[4] L. C. Palacios, *Ingeniería de métodos movimientos y tiempos*. Bogotá: Ecoe Ediciones, 2016.

[5] J. P. Acevedo, "Aplicación de filosofía Lean Manufacturing para optimización de tiempo de ciclo en la industria textil", tesis de maestría, ITESO, 2016. [En línea]. Disponible: https://rei.iteso.mx/items/740d651c-6d10-446e-9809-de1523bde32b

[6] L. C. Arbós, *Procesos en flujo flexible Lean: Organización de la producción y dirección de operaciones*. Madrid, España: Ediciones Díaz de Santos, 2012.

[7] R. Muñoz, "Técnicas de optimización para resolver un problema de programación de producción de una línea de ensamble", tesis doctoral, Universidad Autónoma de Nuevo León, mar. 2018. [En línea]. Disponible: https://rediab.uanl.mx/Record/eprints-16373

[8] R. W. Man, D. Ning y T. H. Tin, "A Mixed-Integer Linear Programming (MILP) for Garment Line Balancing", *IJSRMT*, vol. 4, n.º 2, 2025, doi: 10.5281/zenodo.14942910.

[9] E. G. Vaca, "Balanceo de líneas de producción para la optimización de los procesos productivos en la empresa Alonsso: análisis del proceso productivo de la empresa", trabajo de grado, Escuela Politécnica Nacional, Ecuador, jul. 2025. [En línea]. Disponible: https://bibdigital.epn.edu.ec/handle/15000/26836

[10] J. J. Pachay, "Costos de producción en la empresa ANAMACORP S.A., Cantón Salinas, Provincia de Santa Elena, año 2022", trabajo de grado, jun. 2024, Universidad Estatal Península de Santa Elena, Ecuador, 2024. Accedido: oct. 20, 2025. [En línea]. Disponible: https://repositorio.upse.edu.ec/handle/46000/11639

[11] M. G. Céspedes y W. L. Alcazar, "Mejora continua del proceso de producción aplicando Lean Manufacturing para incrementar la tasa de producción real de una planta embotelladora de agua de mesa", trabajo de grado, Universidad Peruana de Ciencias Aplicadas, nov. 2023. [En línea]. Disponible: https://repositorioacademico.upc.edu.pe/entities/publication/16007fa9-824b-54e3-84ad-a6293406f286

[12] J. C. Seck-Tuoh-Mora, G. E. Anaya-Fuentes, N. Hernández-Romero, J. Medina-Marín, I. Barragan-Vite, and M. A. López-Cabrera, "Optimización de trabajadores y estaciones de trabajo en líneas de ensamble, mediante la adaptación de algoritmos genéticos", *Inter disciplina*, vol. 12, n.º 33, pp. 59-83, 2024. doi: 10.22201/ceiich.24485705e.2024.33.88239.

[13] J. H. Patterson y J. J. Albracht, "Technical Note—Assembly-Line Balancing: Zero-One Programming with Fibonacci Search", *Oper. Res.*, vol. 23, n.º 1, pp. 166-172, feb. 1975, doi: 10.1287/opre.23.1.166.

[14] N. B. Camussi, "Métodos basados en 'slots' para la programación cíclica de líneas de ensamble multiproducto", tesis doctoral, Universidad Nacional del Litoral, Santa Fe, Argentina, 2018. Accedido: oct. 27, 2025. [En línea]. Disponible: https://bibliotecavirtual.unl.edu.ar/handle/11185/5459

[15] Y. Bukchin y T. Raviv, "Constraint programming for solving various assembly line balancing problems", *Omega*, vol. 78, n.º C, pp. 57-68, jul. 2018, doi: 10.1016/J.OMEGA.2017.06.008.

[16] J. D. Mora y J. F. Torres, "Programación lineal entera mixta para el problema de balanceo de línea en U (UALBP-E) minimizando el costo de proceso de las tareas y maximizando la eficiencia de la línea", tesis de maestría, Universidad de los Andes, 2015. Accedido: sept. 8, 2025. [En línea]. Disponible: https://repositorio.uniandes.edu.co/entities/publication/08fc9d30-11bb-4db3-b494-8eefc15d2115

[17] L. Huang *et al.*, "Branch and Bound in Mixed Integer Linear Programming Problems: A Survey of Techniques and Trends", 2021, *arXiv:2111.06257*

[18] Y. Jiao, N. Cao, J. Li, L. Li y X. Deng, "Balancing a U-Shaped Assembly Line with a Heuristic Algorithm Based on a Comprehensive Rank Value", *Sustainability* 2022, vol. 14, n.º 2, p. 775, en. 2022, doi: 10.3390/su14020775.

[19] A. Scholl y S. Voß, "Simple assembly line balancing—Heuristic approaches", *Heuristics*, vol. 2, pp. 217-244, 1997, doi: 10.1007/BF00127358.

[20] P. Saurabh y M. Salman, "An experimental study on the automotive production line using assembly line balancing techniques", *IJMET*, vol. 8, n.º 3, pp. 22-33, mar. 2017. [En línea]. Disponible: https://iaeme.com/MasterAdmin/Journal_uploads/IJMET/VOLUME_8_ISSUE_3/IJMET_08_03_003.pdf

[21] M. T. Çelik y S. Arslankaya, "Solution of the assembly line balancing problem using the rank positional weight method and Kilbridge and Wester heuristics method: An application in the cable industry", *J. Eng. Res.*, vol. 11, n.º 3, pp. 182-191, 2023, doi: 10.1016/j.jer.2023.100082.

[22] O. Battaïa y A. Dolgui, "Hybridizations in line balancing problems: A comprehensive review on new trends and formulations", *Int. J. Prod. Econ.*, vol. 250, p. 108673, 2022, doi: 10.1016/j.ijpe.2022.108673.

[23] J. B. H. C. Didden, E. Lefeber, I. J. B. F. Adan e I. W. F. Panhuijzen, "Genetic algorithm and decision support for assembly line balancing in the automotive industry", *Int. J. Prod. Res.*, vol. 61, n.º 10, pp. 3377-3395, 2023, doi: 10.1080/00207543.2022.2081630.

[24] L. Feng, Y. Wang, X. Fang, H. Yu y S. Zhang, "Two-sided resource-constrained assembly line balancing problem: A new mathematical model and an improved genetic algorithm", *Swarm Evol. Comput.*, vol. 90, p. 101662, 2024, doi: 10.1016/j.swevo.2024.101662.