

UNIVERSIDAD DE LAS AMÉRICAS PUEBLA

ESCUELA DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL Y MECÁNICA

**UDLAP**®

**PROBLEMA DE BALANCEO DE LÍNEAS BILATERALES:  
DESARROLLO DE MÉTODO COMSOAL-2S COMO  
ALTERNATIVA DE SOLUCIÓN**

TESIS QUE, PARA COMPLETAR LOS REQUISITOS DEL PROGRAMA DE  
HONORES PRESENTA LA ESTUDIANTE

MONTSERRAT GUTIÉRREZ BLANCO

ID 167001

DRA. DOLORES EDWIGES LUNA REYES

SAN ANDRÉS CHOLULA, PUEBLA.

OTOÑO, 2023

TESIS QUE, PARA COMPLETAR LOS REQUISITOS DEL PROGRAMA DE  
HONORES PRESENTA LA ESTUDIANTE MONTSERRAT GUTIÉRREZ  
BLANCO ID 167001

DIRECTOR DE TESIS

---

Dra. Dolores Edwiges Luna Reyes

PRESIDENTE DE TESIS

---

Dr. Juan Antonio Díaz García

SECRETARIO DE TESIS

---

Dra. Elsa Velázquez Miranda

# Índice

<b>1. Introducción</b>	<b>4</b>
<b>2. Revisión de la Literatura</b>	<b>10</b>
2.1. Líneas de ensamble . . . . .	10
2.2. Clasificación . . . . .	12
2.3. Problema de balanceo de líneas de ensamble . . . . .	14
<b>3. Metodología</b>	<b>20</b>
3.1. Descripción del problema . . . . .	20
3.2. Suposiciones . . . . .	20
3.3. Modelo matemático . . . . .	21
3.4. COMSOAL-2S . . . . .	25
<b>4. Resultados y Discusión</b>	<b>29</b>
4.1. Modelo matemático . . . . .	29
4.2. COMSOAL-2S . . . . .	31
<b>5. Conclusiones</b>	<b>35</b>
<b>Referencias</b>	<b>38</b>
<b>A. Apéndice A</b>	<b>44</b>

## Resumen

El término *problema de balanceo de líneas de ensamble* hace referencia a la determinación de cómo asignar tareas a las diferentes estaciones de una línea de ensamble, respetando ciertas restricciones y persiguiendo algún objetivo de optimización específico. Este problema presenta distintas variantes en función de las características y configuración de la línea. Una de estas variantes es el problema de balanceo de líneas de ensamble bilaterales (2S-ALBP). Debido a todas las restricciones que deben considerarse, como las relaciones de precedencia entre tareas y lados en que debe hacerse una operación, así como la naturaleza combinatoria del problema, el 2S-ALBP es un problema con un grado de complejidad considerable. Su solución ha sido abordada a través de modelos matemáticos, heurísticas y metaheurísticas. Sin embargo, la cantidad de trabajos en donde se aborda el 2S-ALBP es aún limitada en comparación con la amplia gama enfocada al problema de balanceo de líneas simple (SALBP). Por ende, en la presente investigación se propone un método heurístico, basado en la lógica del método COMSOAL para el SALBP, como alternativa de solución para el 2S-ALBP, y se evalúa su eficiencia aplicándolo a problemas de prueba encontrados en la literatura. Los resultados obtenidos se comparan con aquellos obtenidos empleando un modelo matemático. Los resultados indican que a pesar de su sencillez, el método propuesto proporciona resultados de buena calidad en tiempo de cómputo razonable.

**Palabras clave:** Problema de balanceo de líneas de ensamble, Líneas de ensamble bilaterales, Método heurístico, COMSOAL.

# 1. Introducción

Las líneas de ensamble, desde su concepción y posterior introducción al ámbito de la manufactura a principios del siglo XX como un método novedoso para la fabricación de productos particulares de manera más eficiente, se han convertido en un componente básico de todo tipo de procesos productivos. En los términos más simples, se le denomina línea de ensamble a un conjunto de estaciones de trabajo ordenadas de manera lineal y conectadas a través de uno o varios equipos de manejo de materiales. Así, las piezas ingresan a la línea y van pasando de estación a estación, en donde un operador, que puede ser una persona, maquinaria, o incluso un robot, ejecuta una o varias tareas sobre la pieza antes de dejarla continuar su recorrido hasta la siguiente estación de trabajo (Grzechca, 2013).

Dentro de la definición de líneas de ensamble anteriormente provista, resulta importante destacar la frase "*estaciones de trabajo ordenadas*", pues de ella pueden desglosarse dos de las principales decisiones relacionadas con el diseño e implementación de una línea de ensamble: cómo ordenar las estaciones de trabajo y cómo distribuir entre ellas las tareas que deben llevarse a cabo para tener un producto terminado. En otras palabras, cuál debe ser la configuración de la línea y cómo debe balancearse (Grzechca, 2013).

En primer lugar, la decisión acerca de la configuración de la línea de ensamble depende directamente de lo que exija el proceso de manufactura que va a llevarse a cabo. Así, dependiendo de las características del producto que va a fabricarse, las líneas pueden ser de modelo único, mixtas o multimodelo. Las líneas de modelo único funcionan para la manufactura de grandes cantidades de un solo producto; las líneas mixtas se usan para la fabricación de variantes que difieren ligeramente la una de la otra y parten del mismo

producto base, y las líneas multi modelo se eligen cuando lo que se busca es producir lotes de diferentes productos. También, de acuerdo con el espacio del que se dispone y la distribución que resulte más adecuada, las líneas pueden ser rectas, con forma de U, paralelas o seriales. Además, dependiendo de los lados de operación disponibles, las líneas pueden ser unilaterales o bilaterales (Saif, Guan, Wang, Mirza, y Huang, 2014).

Sobre el tema de lados de operación, es relevante destacar que el uso de líneas de ensamble bilaterales representa ciertas ventajas cuando se le compara contra las unilaterales. Se ha comprobado que una línea bilateral es capaz de producir la misma cantidad de producto y al mismo ritmo que una línea unilateral, pero empleando hasta 33 % menos estaciones de trabajo, por lo que se acorta la línea de ensamble (Bartholdi, 1993). Además, la producción puede llevarse a cabo más rápidamente y disminuyen el manejo de materiales, los costos de herramientas y equipo (Aghajani, Ghodsi, y Javadi, 2014).

En segundo lugar, respecto a la distribución de tareas, la decisión de cómo asignar las tareas a las diferentes estaciones debe llevarse a cabo de acuerdo con una serie de reglas implícitas de la propia secuencia del proceso de producción. Es decir, es indispensable respetar las relaciones de precedencia de las tareas. Asimismo, suele buscarse que la asignación logre alcanzar un objetivo de optimización específico, como minimizar el número de estaciones (tipo I) o minimizar el tiempo de ciclo (tipo II) (Gansterer y Hartl, 2018).

La óptima asignación de tareas a las estaciones de una línea de ensamble es un problema que ha sido ampliamente abordado y estudiado. De hecho, incluso ha sido bautizado con un título distintivo. La expresión en forma matemática de la decisión de cómo asignar tareas a las estaciones de trabajo de una línea, optimizando cierta función objeti-

vo y respetando las restricciones pertinentes, recibe el nombre de problema de balanceo de líneas de ensamble (ALBP, por sus siglas en inglés). Fue formulado por primera vez en 1955, tomando como base el ejemplo más sencillo: una línea de ensamble unilateral de modelo único (Grzechca, 2013). Sin embargo, con el tiempo, se han explorado también los problemas de balanceo para las otras configuraciones de líneas de ensamble posibles (Cerqueus y Delorme, 2019).

Tomando en cuenta lo anterior, el presente estudio se concentra en la resolución del problema de balanceo de líneas para el caso particular de líneas de ensamble bilaterales (2S-ALBP, por sus siglas en inglés), que consiste en asignar las tareas necesarias para la fabricación de un producto determinado al lado derecho o izquierdo de las estaciones de una línea, respetando ciertas restricciones y buscando optimizar uno o varios criterios (Yuwei, Jian'e, Feng, y Shijing, 2020). Esta variante del problema resulta más compleja que el ALBP simple, pues la asignación de tareas a ambos lados de la línea puede verse obstaculizada por las relaciones de precedencia que existan entre ellas (Bartholdi, 1993).

Desde que el problema fue planteado por primera vez, múltiples autores lo han abordado, optando básicamente por dos alternativas de solución: métodos de solución exacta y métodos heurísticos. Si bien existen múltiples modelos propuestos capaces de generar soluciones óptimas, tal como las alternativas planteadas por Wu (2008), Özcan y Toklu (2009), Yadav, Verma y Agrawal (2020), y Kim, Song y Kim (2009), se opta comúnmente por el uso de métodos heurísticos que, aunque hacen posible la obtención de soluciones factibles, permiten un grado de cercanía a la solución óptima variable, como es el caso de los métodos propuestos por Kucukkoc y Zhang (2016), Delice, İlkay, Kızılkaya Aydoğan y Özcan (2017), Hamzadayi (2018), Li, Zhang, Tian, Shao y Li (2018), Yuwei,

Jian'e, Feng y Shijing (2020), y Li, Janardhanan, Tang, Qihua y Nielsen (2021). Esto se debe a que, como el problema de balanceo de líneas de ensamble es considerado un problema de optimización combinatoria tipo NP-hard, es posible que no se puedan encontrar soluciones óptimas, en un tiempo de cómputo razonable, sobre todo en casos donde deba asignarse una gran cantidad de tareas (Grzechca, 2013).

Dentro del amplio abanico de métodos heurísticos existentes, se encuentra el *Computer Method of Sequencing Operations for Assembly Lines* (COMSOAL) que se basa en la generación aleatoria de secuencias y resulta especialmente útil para la asignación de un número grande de tareas (Arcus, 1996). Se ha demostrado su eficiencia en la resolución del problema de balanceo de líneas simple (SALBP), pero, de acuerdo a la literatura consultada, la aplicación de COMSOAL en otras variantes del problema, sobre todo las que implican el uso de líneas de ensamble bilaterales, es limitada (Türkmen, Yesil, y Kayar, 2016).

Sobre la base de la información antes establecida, la presente investigación propone una variante del método heurístico COMSOAL para la resolución de líneas de ensamble bilaterales de modelo único. Además, se estudiará el tipo II del 2S-ALBP, es decir, el objetivo de optimización seleccionado es la minimización del tiempo de ciclo. De aquí en adelante, se emplearán las siglas COMSOAL-2S para hacer referencia al método propuesto.

La selección de la variante de ALBP que será abordada en el presente proyecto se basa fundamentalmente en cuatro puntos. En primer lugar, las líneas bilaterales tienen importantes aplicaciones en la industria, sobre todo para la manufactura de productos de mayor tamaño. En segundo lugar, su uso representa una serie de beneficios frente a la

utilización de líneas unilaterales, incluyendo una reducción en movimientos y manejo de materiales, una disminución en tiempos de procesamiento y tiempos de preparación, así como un menor costo de herramental (la misma herramienta puede ser compartida por estaciones a lados opuestos de la línea). Además, el hecho de que exista más de un lado posible de operación representa un desafío adicional al momento de asignar tareas y puede afectar la productividad de la línea, por lo que resulta relevante explorar distintas alternativas de solución (Grzechca, 2013). Por último, las líneas unilaterales ya han sido ampliamente estudiadas para el problema de balanceo, pero a las líneas bilaterales se les ha dado mucha menos atención (Purnomo, Wee, y Rau, 2013).

Por otra parte, la decisión del método de solución abordado se basa principalmente en la naturaleza del problema. En primer lugar, se ha demostrado que el balanceo de líneas es un problema de tipo NP-Hard, lo cual implica que no se resuelve de manera eficiente en tiempo polinomial (Janardhanan, Li, y Nielsen, 2019). Por esta razón, se requieren algoritmos eficientes, diseñados teniendo en mente los requerimientos específicos del ALBP, para poder generar soluciones de calidad en corto tiempo, que puedan implementarse para el balanceo de líneas de ensamble (Yin, Luo, Sun, y Zhang, 2021). Para la solución de problemas de optimización combinatoria de este tipo, aunque sí existen algoritmos exactos que obtienen resultados adecuados en tiempos de solución admisibles para problemas medianos, los métodos heurísticos resultan más prometedores, sobre todo a medida que la cantidad de tareas por asignar aumenta (Pérez O. y cols., 2004).

Considerando lo anterior, el propósito de este proyecto tiene tres partes. En primer lugar, proponer un método heurístico basado en el algoritmo COMSOAL para la resolución del 2S-ALBP, que permita obtener soluciones factibles para el 2S-ALBP. Este

heurístico se implementa utilizando el software *FICO Xpress Optimization*. Asimismo, implementar un modelo de solución exacta para el 2S-ALBP, de manera que pueda resolverse a través del software *FICO Xpress Optimization*, para obtener resultados que permitan tener un punto de partida desde donde se pueda analizar la efectividad del modelo COMSOAL-2S propuesto. Finalmente, mostrar una comparación de los resultados que arrojan ambas alternativas, con el fin de contrastarlos y poder evaluar la efectividad del método propuesto y establecer conclusiones.

Lo que resta de este documento se divide en cuatro capítulos. En la primera sección, *Revisión de la literatura*, se presenta un marco teórico sobre el problema de balanceo de líneas, así como los principales autores que lo han abordado y el enfoque que eligieron. En el segundo apartado, *Metodología*, se detallan la descripción del 2S-ALBP y las suposiciones consideradas, al igual que la formulación y notación del modelo utilizado y el método heurístico desarrollado. Posteriormente, se presenta la sección de *Resultados y Discusión*, donde se exponen los hallazgos más importantes y se analizan de manera más profunda. Finalmente, en el último capítulo se exponen las conclusiones finales del trabajo.

## 2. Revisión de la Literatura

### 2.1. Líneas de ensamble

Desde comienzos del siglo XX, cuando fueron introducidas por Ford Motor Company para la producción de automóviles, las líneas de ensamble han fungido un rol clave en el ámbito industrial (Y. Li, 2017). Una línea de ensamble es un método de fabricación en el que un conjunto de productos o piezas pasan a través de una serie de estaciones de trabajo, donde se procesan hasta obtener un producto terminado (Pereira, 2018).

A pesar de que en un inicio surgió como una iniciativa de la industria automotriz, actualmente este método de manufactura se utiliza en diversas industrias, con el objetivo de fabricar grandes cantidades de productos estandarizados. Así, las líneas de ensamble permiten satisfacer las demandas del mercado con piezas de alta calidad y bajo coste (Delice y cols., 2017).

La estructura básica de una línea de ensamble consiste en una serie de trabajadores encargados de armar un producto, que va avanzando sobre un dispositivo móvil, como una cinta transportadora, siguiendo una secuencia de tareas (Y. Li, 2017). De acuerdo con Özcan y Toklu (2010), en este contexto, una tarea se define como un elemento de trabajo indivisible. El tiempo requerido para llevar a cabo una tarea se conoce como tiempo de tarea, que puede ser determinístico o estocástico (Pınarbaşı, Yüzükırmızı, y Toklu, 2016).

El concepto de tiempo de tarea determinístico se refiere a la suposición de que maquinaria avanzada, o bien, operarios altamente calificados, llevan a cabo sus tareas en un tiempo constante. Por otro lado, al hablar de tiempos estocásticos, se considera que, a

causa de diversas situaciones, como averías en el equipo, curvas de aprendizaje por parte de los empleados y errores en tareas complejas, el tiempo que cada tarea requiere para llevarse a cabo puede variar de pieza a pieza (Pınarbaşı y cols., 2016).

En una línea de ensamble, se le conoce como estación de trabajo a un segmento de la línea en el que se llevan a cabo operaciones. A cada estación le corresponde la ejecución de un grupo de tareas, que se realizan por un operador. Según la naturaleza de la línea de producción, el operador puede ser una máquina o una persona (Pınarbaşı y cols., 2016). Cada una de las estaciones opera con un tiempo predeterminado y la suma del tiempo requerido para completar las tareas en cada una de las estaciones de trabajo que componen el proceso productivo se conoce como tiempo de ciclo (Hamzadayı, 2018). El tiempo en el que se llevan a cabo las tareas asignadas a cada estación no puede superar el tiempo de ciclo de la línea, por lo que este último determina el ritmo de producción de una línea o sistema (Kucukkoc y Zhang, 2016).

Resulta fundamental aclarar que el tiempo de ciclo de la línea es distinto del tiempo de ciclo en cada estación. La diferencia radica en que el tiempo de ciclo de la línea de ensamble es el tiempo de ciclo máximo de todas las estaciones de trabajo. En otras palabras, el tiempo de ciclo de la línea de ensamble indica el tiempo que pasa desde que un producto terminado sale de la línea de producción hasta que sale el siguiente producto terminado. Por otro lado, el tiempo de ciclo en cada estación se refiere al tiempo de operación de estas; es decir, es el tiempo requerido para llevar a cabo todas las tareas asignadas a una estación dada, y puede variar de estación a estación (Sivasankaran y Shahabudeen, 2014).

Normalmente, las diferentes estaciones de trabajo se comunican a través de un siste-

ma de transporte que traslada los productos a lo largo de la línea de ensamble (Kucukkoc y Zhang, 2016). Es importante mencionar que resulta crucial que se terminen las tareas asignadas en una estación antes de que el producto pueda pasar a la siguiente. Es decir, si la fabricación de un producto requiere que se lleven a cabo las tareas A y B, asignadas respectivamente a las estaciones 1 y 2, entonces el producto comenzará en la estación 1, donde se realizará la tarea A, y no pasará a la estación 2 hasta que esta se haya terminado y pueda empezarse con la tarea B. En este ejemplo, la tarea A precede a la tarea B. Esto se conoce como relación de precedencia y debe tomarse en cuenta al asignar tareas a las distintas estaciones (Lopes, Sikora, Michels, y Magatão, 2020).

## **2.2. Clasificación**

A pesar de que todas las líneas de producción siguen los mismos principios básicos, con el tiempo se han propuesto distintas configuraciones y se han implementado modificaciones para facilitar el proceso de fabricación. Con base en lo anterior, las líneas de ensamble pueden clasificarse según el tipo de flujo, el número de lados de operación y la variedad de productos ensamblados, entre otros.

De acuerdo con el tipo de control de flujo, las líneas de ensamble pueden ser continuas, síncronas o asíncronas. Las líneas continuas cuentan con una cinta transportadora que, constantemente, mueve todas las piezas. En las líneas síncronas, todos los productos se mueven juntos, pero pasan a su siguiente estación de trabajo únicamente cuando todas las piezas han sido procesadas en la estación actual. Por último, en las líneas asíncronas, cada pieza pasa de estación a estación en forma independiente, siempre y cuando ya hayan sido procesadas y la siguiente estación esté disponible (Lopes y cols., 2020).

Considerando el número de lados de operación, las líneas de producción pueden clasificarse como unilaterales o bilaterales. La unilateralidad, que es cuando solamente se opera en uno de los lados de la línea, ya sea el derecho o el izquierdo, es la configuración tradicional (Hamzadayı, 2018). Por otro lado, en las líneas de ensamble bilaterales pueden llevarse a cabo tareas en cualquiera de los lados o en ambos lados simultáneamente. En este tipo de líneas, cada estación se representa por un par de estaciones, conocidas como estaciones de trabajo acopladas, que se encuentran en lados opuestos de la línea de ensamble y los operarios trabajan en paralelo, llevando a cabo distintas tareas sobre una misma pieza (Kucukkoc y Zhang, 2016). Esta configuración fue introducida por Bartholdi en 1993 y es comúnmente utilizada en la manufactura de productos de gran tamaño, como camionetas, camiones o automóviles (Kucukkoc y Zhang, 2014). En comparación con las líneas unilaterales, las líneas de ensamble bilaterales presentan ciertas ventajas. En primer lugar, se acorta la línea de ensamble. Además, la producción puede llevarse a cabo en menos tiempo, pues hay tareas que se realizan a la vez. También, disminuye el manejo de material, así como los costos de herramientas y equipo (Aghajani y cols., 2014).

Finalmente, tomando en cuenta la variedad de modelos que se fabrican en la misma línea, las líneas de ensamble pueden clasificarse como líneas de modelo único, líneas multimodelo y líneas mixtas. Cuando se produce una gran cantidad de un mismo tipo de producto, se usan líneas de modelo único (Kucukkoc y Zhang, 2014). La desventaja de este tipo de líneas es que no son convenientes para la manufactura de productos que requieren adecuarse a demandas personalizadas por el cliente, ya que son bastante inflexibles (Wilson, 2014). Por esta razón, comenzaron a implementarse en la industria

líneas multimodelo y mixtas. En las líneas multimodelo, es posible producir lotes de distintos modelos de productos. Sin embargo, su uso presenta también ciertas desventajas, ya que requieren de tiempo destinado a operaciones de reconfiguración de la línea de producción entre cada lote (Kucukkoc y Zhang, 2014). En las líneas mixtas también pueden producirse varios modelos del mismo producto básico, pero, a diferencia del caso anterior, el tiempo requerido para operaciones de reconfiguración es mínimo y, por lo tanto, despreciable (Lopes y cols., 2020). Este tipo de líneas de ensamble es altamente empleado en las industrias automotriz y electrónica, caracterizadas por alta variabilidad en la demanda y un menor volumen de producción (C. Yang, Gao, y Jinlin, 2014). Una de las ventajas que el uso de líneas mixtas conlleva es que, en vez de invertir en la construcción y mantenimiento de líneas independientes para cada modelo que se produce, los fabricantes pueden obtener el mismo resultado a partir de una única línea (Taha, El-Kharbotly, Afia, y Sadek, 2012). Además, la producción mixta es capaz de satisfacer demanda variable al incrementar la flexibilidad, así como reducir costos de inventario (Aghajani y cols., 2014).

### **2.3. Problema de balanceo de líneas de ensamble**

A pesar de sus diferencias, todas las configuraciones y variantes de las líneas de ensamble deben balancearse. Este problema consiste en determinar la asignación de tareas a las diferentes estaciones de trabajo, de acuerdo a cierto objetivo, mientras se satisfacen relaciones de precedencia y otras restricciones específicas (Pinarbaşı y cols., 2016). El ALBP es una decisión crucial en el diseño y manejo de las líneas de ensamble, pues su eficiencia está ligada a la forma en que las tareas se reparten entre las estaciones y cómo

esto afecta el tiempo de ciclo y la productividad (C. Yang y cols., 2014).

Debido a las diferentes condiciones que deben considerarse y a su complejidad, se ha demostrado que el balanceo de líneas es un problema de tipo NP-Hard (Janardhanan y cols., 2019). Por esta razón, se requieren algoritmos eficientes, diseñados teniendo en mente los requerimientos específicos del ALBP, para poder generar soluciones de calidad en un tiempo de cómputo razonable (Yin y cols., 2021).

El ALBP fue descrito y formulado en primera instancia por Salveson en 1955. A partir de ahí, han surgido distintas variantes del problema básico que consideran diferentes configuraciones de las líneas de ensamble y persiguen diversos objetivos (Cerqueus y Delorme, 2019). Entre las principales variantes destacan el problema simple de balanceo de línea (SALBP), el general (GALBP) y el mixto (MALB) (Pereira, 2018).

En el SALBP, las diferentes tareas, cada una de ellas con tiempos determinísticos, se asignan a las estaciones de trabajo respetando sus relaciones de precedencia, buscando optimizar la eficiencia de la línea. Dependiendo del objetivo particular que se persigue, pueden distinguirse cuatro variantes del problema simple de balanceo de línea: tipo I, tipo II, tipo F y tipo E (Y. Li, 2017). En problemas de tipo I, se busca minimizar el número de estaciones de trabajo, a partir de un tiempo de ciclo dado. En los problemas de tipo II, se busca minimizar el tiempo de ciclo mediante la asignación de tareas a un número dado de estaciones de trabajo. En los problemas tipo F, se busca lograr un balanceo factible para un número dado de estaciones de trabajo con un tiempo de ciclo establecido. Por último, en los problemas de tipo E, el objetivo es maximizar la eficiencia de la línea y es posible manipular tanto el número de estaciones de trabajo como el tiempo de ciclo (Gansterer y Hartl, 2018).

El problema general de balanceo de líneas de ensamble surge a partir de una generalización del problema simple. La principal distinción es que, en el GALBP, se toman en cuenta consideraciones adicionales para adaptar los objetivos y restricciones a los requerimientos de problemas reales (Pereira, 2018).

Finalmente, los problemas de balanceo de líneas de ensamble mixtas (MALBP) consideran que hay varios modelos de cierto producto, donde cada uno tiene un conjunto de tareas y tiempos de procesamiento. Así, debe determinarse cómo asignar las tareas a las diferentes estaciones, respetando las relaciones de precedencia y tomando en cuenta las diferencias de los modelos (Aghajani y cols., 2014).

En todas estas variantes del problema podemos considerar líneas de ensamble bilaterales (Yuwei y cols., 2020), que es la cuestión particular que aborda el presente proyecto. La línea de ensamble bilateral se diseñó con el objetivo de fabricar productos de gran tamaño (D. Li y cols., 2018). Al utilizar esta configuración de línea, se obtienen algunas ventajas como menor longitud de línea, mayor aprovechamiento de herramientas y equipo y mayor flexibilidad, lo que permite una mejor adaptación a la demanda del mercado (Delice y cols., 2017). Asimismo, se aprovecha el espacio de manera más eficiente y se reducen costos de transporte, manejo de materiales y tiempo de reconfiguración (Yadav y cols., 2020). En este tipo de líneas, cada estación acoplada cuenta con dos operarios que trabajan en lados opuestos, llevando a cabo las tareas requeridas para cada modelo del producto. Para balancear la línea, las tareas deben clasificarse como tareas de lado izquierdo (L), tareas de lado derecho (D) o tareas que pueden hacerse en cualquiera de los lados de la línea (E) (Taha y cols., 2012).

Al igual que en otros casos, la asignación de tareas debe respetar relaciones de pre-

cedencia. No obstante, este caso implica mayor complejidad, pues deben determinarse también relaciones de precedencia dentro de cada estación. Por ejemplo, si se tiene una tarea  $i$ , que precede a una tarea  $j$  y ambas están asignadas a la misma estación acoplada, entonces  $j$  no puede empezar hasta que  $i$  sea concluida (Gansterer y Hartl, 2018). Además, debido a la configuración de la línea de ensamble, deben considerarse restricciones adicionales, como restricciones de zonificación y posicionamiento, dado que ciertas tareas deben estar en una estación específica o en la misma estación acoplada por cuestiones de uso de herramientas y equipo (D. Li y cols., 2018).

El primer estudio sobre 2S-ALBP fue realizado en 1993 por Bartholdi, quien propuso un programa interactivo que puede utilizarse para la asignación de tareas, así como una serie de reglas de asignación que pueden emplearse para la resolución de este tipo de problemas (Lee, Kim, y Kim, 2001).

Desde entonces, el problema ha sido abordado por otros autores. En algunos casos, se han utilizado métodos de solución exacta. Wu (2008) plantea el uso de algoritmos de ramificación y acotamiento. Özcan y Toklu (2009) proponen modelos de optimización multiobjetivo (Kang y Lee, 2023). De igual forma, Yadav, Verma y Agrawal plantean un modelo matemático mixto para maximizar la carga de trabajo en cada estación y minimizar el tiempo de inactividad (Yadav y cols., 2020). Similarmente, Kim, Song y Kim proponen un modelo matemático para la resolución del 2S-ALBP, que además puede utilizarse para desarrollos futuros (Kim y cols., 2009).

Por otra parte, destaca también el uso de heurísticas y metaheurísticas. En primer lugar, Kucukkoc y Zhang integran el algoritmo genético y el de colonia de hormigas para resolver el problema y determinar la secuenciación de los productos fabricados

en la línea (Kucukkoc y Zhang, 2016). Asimismo, Delice, İlkay, Kızılkaya Aydoğan y Özcan utilizan un algoritmo de optimización de enjambre de partículas como estrategia de solución (Delice y cols., 2017). Por otro lado, Hamzadayi plantea el uso de algoritmos enseñanza-aprendizaje para minimizar el número de estaciones y estaciones acopladas (Hamzadayı, 2018). También, Li, Zhang, Tian, Shao y Li introducen un algoritmo competitivo imperialista para abordar un problema de balanceo multiobjetivo (D. Li y cols., 2018). Similarmente, Yuwei, Jian'e, Feng y Shijing proponen un planteamiento con múltiples restricciones, buscando usar un algoritmo genético para minimizar el número de estaciones y el índice de suavidad, el cual se relaciona con la suavidad del flujo de la línea, donde un valor de cero indicaría una línea perfectamente balanceada (Yuwei y cols., 2020). Además, Li, Janardhanan, Tang y Nielsen presentan el uso de algoritmos de búsqueda local para solucionar problemas 2S-ALBP tipo I (Z. Li y cols., 2021).

Los métodos heurísticos y metaheurísticos mencionados anteriormente, a pesar de resultar útiles en la resolución del 2S-ALBP, no fueron diseñados únicamente con esta aplicación en mente. Por el contrario, existe un método heurístico llamado Computer Method of Sequencing Operations for Assembly Lines (COMSOAL) que, como su nombre indica, es un procedimiento diseñado específicamente para la asignación de tareas en líneas de ensamble. El modelo emplea un método de generación de secuencias y resulta especialmente útil para procesos con un gran número de tareas (Arcus, 1996). A pesar de que se ha demostrado su eficiencia en la resolución del ALBP, la aplicación de COMSOAL en otras variantes del problema es limitada (Türkmen y cols., 2016).

Es importante destacar que, aunque el problema de balanceo de líneas de ensamble

general ha sido ampliamente estudiado, el número de estudios sobre el problema de balanceo de líneas bilaterales es aún mucho menor (Delice y cols., 2017). Por ende, considerando las diversas aplicaciones y las ventajas que este tipo de líneas ofrece, la investigación sobre el 2S-ALBP resulta relevante.

### **3. Metodología**

#### **3.1. Descripción del problema**

El problema de balanceo de líneas bilaterales básico (2S-ALBP), objeto de estudio de la presente investigación, consiste en la asignación de tareas a los lados izquierdo y derecho de una línea de ensamble con estaciones de trabajo acopladas, respetando las relaciones de precedencia de las tareas. Considerando que se decidió trabajar con el 2S-ALBP tipo II, el objetivo del problema es minimizar el tiempo de ciclo de la línea de ensamble, partiendo de un número de estaciones dado (Yuwei y cols., 2020).

#### **3.2. Suposiciones**

En esta investigación se aborda el problema de balanceo de líneas bilaterales básico con objetivo tipo II, es decir, minimización del tiempo de ciclo con número de estaciones fijo (Özcan y Toklu, 2009). Con base en lo anterior, para los métodos de solución empleados se consideran las siguientes suposiciones:

1. Los tiempos de tarea son conocidos, determinísticos e independientes de la estación en que se asignan las tareas.
2. Las relaciones de precedencia entre las tareas se conocen.
3. Se fabrica un único modelo de un producto en la línea de ensamble.
4. No hay restricciones en cuanto a equipamiento: cualquier tarea puede asignarse a cualquier estación, siempre y cuando se respeten los lados en que debe hacerse la tarea y las relaciones de precedencia.

5. Las tareas asignadas al lado izquierdo y al lado derecho de una estación acoplada se llevan a cabo en paralelo, sin sobrepasar el tiempo de ciclo.

### 3.3. Modelo matemático

En esta sección mostramos el modelo matemático propuesto por Kim y cols. (2009), que es el utilizado en este trabajo. A continuación, se presenta la notación y la formulación del modelo.

#### Notación

$I =$  conjunto de tareas,  $|I| = n$

$J =$  conjunto de estaciones acopladas,  $|J| = m$

$KK =$  conjunto de lados posibles de operación,  $|KK| = 1, 2$

$K =$  lado preferido de operación de la tarea  $i$ . El número  $1$  indica lado izquierdo y el número  $2$ , lado derecho

$AL =$  conjunto de tareas que deben realizarse del lado izquierdo de la línea de ensamble,

$$AL \subset I$$

$AR =$  conjunto de tareas que deben realizarse del lado derecho de la línea de ensamble,

$$AR \subset I$$

$AE =$  conjunto de tareas que pueden realizarse en ambos lados de la línea de ensamble,

$$AE \subset I$$

$P_i =$  conjunto de los predecesores inmediatos de la tarea  $i$

$PA_i =$  conjunto de todos los predecesores de la tarea  $i$

$S_i =$  conjunto de los sucesores inmediatos de la tarea  $i$

$SA_i =$  conjunto de todos los sucesores de la tarea  $i$

$P0 =$  conjunto de tareas que no tienen ningún predecesor inmediato,  $P0 \subset I$

$t_i =$  tiempo de procesamiento de la tarea  $i$

$C =$  conjunto de tareas cuyo lado preferido de operación es opuesto al de la tarea  $i$

$u =$  número entero grande

## Variables

$x_{ijk} =$  1, si la tarea  $i$  se asigna a la estación  $j$  del lado  $k$ ,  $\forall i \in I, j \in J, k \in KK$

0, en otro caso.

$z_{ip} =$  1, si la tarea  $i$  se asigna antes que la tarea  $p$  en la misma estación,  $\forall i \in I, p \in P_i$

0, en otro caso.

$tf_i =$  tiempo de terminación de la tarea  $i$   $\forall i \in I$

$ct =$  tiempo de ciclo de la línea

## Modelación

$$\text{F.O. } \min z = ct \quad (1)$$

$$\text{s.a. } \sum_{j \in J} \sum_{k \in K(i)} x_{ijk} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{g \in J} \sum_{k \in K(h)} g * x_{hgk} \leq \sum_{j \in J} \sum_{k \in K(i)} j * x_{ijk} \quad \forall i \in I | i \notin P0, h \in P_i \quad (3)$$

$$tf_i \leq ct \quad \forall i \in I \quad (4)$$

$$tf_i - tf_h + u * (1 - \sum_{k \in K(h)} x_{hjk}) + u * (1 - \sum_{k \in K(i)} x_{ijk}) \geq t_i \quad (5)$$

$$\forall i \in I | i \notin P0, h \in P_i, j \in J \quad (5)$$

$$tf_p - tf_i + u * (1 - x_{pjk}) + u * (1 - x_{ijk}) + u * (1 - z_{ip}) \geq t_p$$

$$\forall i \in I, p \in r | r \in I - (PA(i) \cup SA(i) \cup C(i)), j \in J, k \in K(i) \cap K(p) \quad (6)$$

$$tf_i - tf_p + u * (1 - x_{pjk}) + u * (1 - x_{ijk}) + u * z_{ip} \geq t_i$$

$$\forall i \in I, p \in r | r \in I - (PA(i) \cup SA(i) \cup C(i)), j \in J, k \in K(i) \cap K(p) \quad (7)$$

$$tf_i \geq t_i \quad \forall i \in I \quad (8)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in KK \quad (9)$$

$$z_{ip} \in \{0, 1\} \quad \forall i \in I, p \in P_i \quad (10)$$

$$tf_i \geq 0 \quad \forall i \in I \quad (11)$$

$$ct \geq 0 \quad (12)$$

El modelo matemático previamente descrito emplea diversos conjuntos de restricciones, cuya función se describe a continuación. Algunos de ellos, los conjuntos (2), (3) y (4) para ser exactos, se encuentran con frecuencia en modelos diseñados para el problema de balanceo de líneas, mientras que otros fueron aportados por los autores del

modelo como parte de la contribución de su estudio. El conjunto de restricciones (2) se establece para asegurar que cada tarea sea asignada únicamente a una estación. Este tipo de restricciones son normalmente denominadas restricciones de asignación. Por su parte, el conjunto de restricciones (3) se implementa para cerciorar que se respeten las relaciones de precedencia de las tareas al llevar a cabo la asignación de estas. Adicionalmente, se utiliza el conjunto de restricciones (4) para evitar que el tiempo de terminación de cualquier tarea supere al tiempo de ciclo, ya que es precisamente el tiempo de ciclo lo que determina el ritmo de producción de una línea y, justamente por esta razón, el tiempo en el que se llevan a cabo las tareas asignadas a cada estación no debe superarlo (Kim y cols., 2009).

El conjunto de restricciones (5) se activa cuando dos tareas  $i$  y  $h$  se asignan a la misma estación acoplada  $j$ , pero únicamente si la tarea  $h$  es predecesor inmediato de la tarea  $i$ . Al cumplirse esta condición, la restricción permite que la tarea  $i$  se inicie de manera inmediata una vez que se haya terminado la tarea  $h$ . En el caso de que la condición no se cumpla porque las tareas se asignan en diferentes estaciones, la restricción no obliga a que se inicie  $i$  inmediatamente después de  $h$  (Kim y cols., 2009).

Por otro lado, los conjuntos de restricciones (6) y (7) conciernen a las tareas que no presentan relaciones de precedencia entre ellas. A pesar de que dos tareas  $i$  y  $p$  no tengan una relación de precedencia establecida, si se asignan a la misma estación una debe iniciarse inmediatamente después de que la otra termine, de acuerdo con la secuencia en la que hayan sido asignadas. Si  $i$  se asigna antes de  $p$ , el conjunto de restricciones (6) se activa para permitir que  $p$  inicie inmediatamente después de  $i$ , siempre y cuando ambas tareas hayan sido asignadas a la misma estación acoplada, del mismo lado de la línea.

En el caso contrario, si  $p$  se asigna antes de  $i$ , el conjunto de restricciones (7) se activa para permitir que  $i$  inicie inmediatamente después de  $p$ , siempre y cuando ambas tareas hayan sido asignadas a la misma estación acoplada, del mismo lado de la línea (Kim y cols., 2009).

Asimismo, el conjunto de restricciones (8) se emplea para asegurar que se respete el tiempo de procesamiento de cada tarea, de manera que una tarea nunca pueda concluirse más rápido de lo dictado por su tiempo de procesamiento. Finalmente, en los conjuntos (9), (10), (11) y (12) se establecen también restricciones en cuanto a la naturaleza de las variables, de acuerdo con lo que se define en la sección de *Notación*. Por ende, se establece que las variables  $x$  y  $z$  deben ser binarias, mientras que  $tf$  y  $ct$  solo pueden tomar valores positivos (Kim y cols., 2009).

### **3.4. COMSOAL-2S**

En el caso del algoritmo COMSOAL-2S, se tomó como referencia el algoritmo heurístico propuesto por Nkasu y Leung (1995) para el problema de balanceo de líneas unilaterales (SALBP). El algoritmo se basa en la generación aleatoria de secuencias para la asignación de tareas. Este algoritmo se adapta debido a que el algoritmo original está diseñado para resolver problemas tipo I y en este trabajo se aborda un problema tipo II.

Recordemos que los problemas tipo I tienen como objetivo minimizar el número de estaciones, dado un tiempo de ciclo  $C$ , mientras que los problemas tipo II tienen como objetivo minimizar el tiempo de ciclo, dado un número de estaciones. Con esto en mente, la adaptación del COMSOAL consiste en resolver inicialmente el problema con el tiempo de ciclo mínimo posible y revisar el número de estaciones de trabajo que se obtienen

con este tiempo de ciclo. En caso de que el número de estaciones exceda al número de estaciones deseado, el tiempo de ciclo se aumenta en una unidad y se ejecuta nuevamente el algoritmo COMSOAL. Este proceso se repite hasta que el algoritmo obtiene una solución con el número de estaciones deseadas (Ver Algoritmo 1).

### **Notación**

$N =$  número de tareas

$C =$  tiempo de ciclo

$X =$  número máximo de iteraciones

$UB =$  cota superior

$IDLE =$  tiempo de ocio de la línea

$t_i =$  tiempo de procesamiento de la tarea  $i$

$A =$  conjunto de tareas por asignar

$B =$  conjunto de tareas por asignar cuyos predecesores inmediatos ya han sido asignados

$F =$  conjunto de tareas factibles para asignación

$cL =$  tiempo disponible para asignar en la estación activa del lado izquierdo

$cR =$  tiempo disponible para asignar en la estación activa del lado derecho

---

**Algorithm 1** Algoritmo COMSOAL-2S
 

---

Sea  $Obj$  el número de estaciones acopladas deseadas,  $C$  el tiempo de ciclo y  $X$  el número

de soluciones a generar.

$$C := \left\lceil \frac{\sum_{i=1}^N t_i}{Obj} \right\rceil - 1$$

**repeat**

$$C := C + 1$$

Ejecutar *COMSOALAd*

**until**  $NEst - Obj = 0$

---

En el algoritmo COMSOALAd (ver Algoritmo 2), se generan en forma aleatoria soluciones factibles para el problema. Para generar cada solución, el algoritmo utiliza tres conjuntos. El conjunto  $A$ , que contiene las tareas que no han sido asignadas, el conjunto  $B$ , que es un subconjunto de  $A$  que contiene las tareas que satisfacen las restricciones de precedencia, y el conjunto  $F$ , que es el subconjunto de  $B$  formado por las tareas que se pueden asignar en la estación abierta debido a que sus tiempos de proceso no exceden el tiempo que aún está disponible en la estación abierta.

Cada solución se construye de manera iterativa. En cada iteración, se selecciona una tarea para añadirse a la estación abierta. También, en cada iteración se actualizan los conjuntos  $B$  y  $F$ . Si  $F$  es diferente al conjunto vacío, se selecciona una tarea de ese conjunto y se añade a la solución. En caso de que  $F$  sea el conjunto vacío, significa que ya no hay tareas que puedan añadirse a la estación abierta, por lo que debe abrirse una nueva estación. Una vez que una solución se completa, se compara contra la incumbente (la mejor solución obtenida hasta el momento) y en caso de que esta solución sea mejor,

se actualiza la incumbente.

La asignación de la tarea seleccionada se hace respetando el lado en que la operación debe realizarse. Para las operaciones que se pueden hacer en cualquiera de los dos lados, se hace la asignación en el lado en el que queda mayor tiempo de ocio después de haber asignado la tarea.

---

**Algorithm 2** COMSOALAd
 

---

```

UB := ∞
for x = 1 to X do
  NE := 1
  A := {1 . . . N}
  cR := C, cL := C, IDLE := 0
  while A ≠ {} do
    Formar el conjunto B con todas las tareas de A que cumplen sus precedencias
    Formar el conjunto F con todas las tareas de B que no exceden el tiempo disponible
    en la estación abierta
    if F ≠ {} then
      Seleccionar aleatoriamente i* de F y asignarla a la estación abierta
      Actualizar las tareas que cumplen precedencias
      Actualizar cR, cL
      A := A - {i*}
    else
      Abrir nueva estación
      NE := NE + 1
      IDLE := IDLE + cR + cL
      if IDLE > UB then
        break
      end if
    end if
  end while
  IDLE := IDLE + cR + cL
  if IDLE < UB then
    NEst := NE
    UB := IDLE
  end if
end for
  
```

---

## 4. Resultados y Discusión

Para evaluar la calidad de las soluciones obtenidas con el algoritmo propuesto en este trabajo, COMSOAL-2S, se realizaron una serie de pruebas en las que se utilizaron 22 instancias de prueba, con diferentes números de tareas y estaciones. Estas instancias fueron obtenidas de literatura sobre el tema. Las instancias 1, 2, 7, 8 y 9 (9 y 12 tareas) se obtuvieron de Kim y cols. (2000); la número 3 (16 tareas), de Papadopoulos y cols. (2021); los problemas 4, 5 y 6 (24 tareas) de Z. Yang y cols. (2016), y los ejemplos 10, 11, 12, 13 y 14 (65 tareas) fueron obtenidos de Lee y cols. (2001), al igual que las instancias 15 a 22 (205 tareas). Todas estas instancias se muestran con mayor detalle en el *Apéndice A*. Tanto el modelo matemático como el algoritmo propuesto se codificaron en *FICO Xpress Optimization* y se corrieron en una computadora con un procesador Intel(R) Xeon(R) E-2176M, con un CPU @ 2.70GHz y 32GB de memoria RAM. A continuación se presentan los resultados obtenidos.

### 4.1. Modelo matemático

En primer lugar, para tener un punto de referencia para evaluar los resultados del método heurístico, cada uno de los problemas de prueba se resolvió con el modelo matemático, empleando el software *FICO Xpress Optimization* y con una hora como tiempo límite. En la *Tabla 1*, se muestran los resultados obtenidos en esta prueba. La primera columna de la tabla muestra el número de instancia. La segunda y tercera columnas presentan la cota inferior y superior obtenidas por el software. La cuarta columna muestra el GAP porcentual y finalmente, la última columna muestra el tiempo que le tomó al

software obtener los resultados. Para aquellas instancias en donde el tiempo es de 3600 segundos, indica que no fue posible obtener la solución óptima en el tiempo límite establecido.

Como puede observarse en la *Tabla 1*, en la mayor parte de las instancias (14 de las 22 instancias), el modelo no pudo encontrar la solución óptima en una hora. Podemos también observar que aún para algunas instancias pequeñas como la instancia 6, de sólo 24 tareas, el modelo no puede cerrar el GAP en el tiempo límite establecido. En promedio el GAP porcentual es de 45.56 %. Como era de esperarse este GAP se hace más pronunciado para las instancias de mayor número de tareas. El hecho de que el desempeño del modelo decline a medida que aumenta el número de tareas limita sus aplicaciones, ya que en situaciones reales, sobre todo considerando los tipos de productos que se fabrican en líneas de ensamble bilaterales, como vehículos y otros productos de gran tamaño, es probable que se tengan mayores números de tareas.

Las soluciones obtenidas y los tiempos de ejecución no resultan del todo inesperados. De hecho, lo ocurrido ejemplifica una situación común al intentar aplicar métodos de solución exacta al problema de balanceo de líneas pues, debido a su naturaleza, y sobre todo en el caso del 2S-ALBP dados los desafíos adicionales que representa, simplemente existen casos en los que la complejidad del problema no permite encontrar una solución óptima y se requieren largos tiempos de ejecución. Es por esto precisamente que varios autores suelen optar por métodos de solución alternativos, como heurísticas y metaheurísticas, que han comprobado resultar más eficientes.

**Tabla 1**

*Resultados obtenidos para el tiempo de ciclo con el modelo matemático*

Instancia	Cota inferior	Cota superior	%GAP	Tiempo (s)
1	5	5	0.00 %	0
2	7	7	0.00 %	0.2
3	22	22	0.00 %	0.1
4	18	18	0.00 %	3.9
5	24	24	0.00 %	29.9
6	34	35	2.94 %	3600
7	3	3	0.00 %	0
8	5	5	0.00 %	0
9	4	4	0.00 %	0.1
10	338	666	97.04 %	3600
11	313	532	69.97 %	3600
12	272	447	64.34 %	3600
13	272	383	40.81 %	3600
14	272	336	23.53 %	3600
15	1455	2725	87.29 %	3600
16	944	2446	159.11 %	3600
17	944	2180	130.93 %	3600
18	944	1877	98.83 %	3600
19	944	1724	82.63 %	3600
20	944	1535	62.61 %	3600
21	944	1205	27.65 %	3600
22	944	1460	54.66 %	3600

## 4.2. COMSOAL-2S

Para evaluar el comportamiento del método propuesto, cada una de las instancias de prueba se ejecutó con el algoritmo COMSOAL-2S, empleando el software *FICO Xpress Optimization*. En este caso, como el modelo contiene elementos aleatorios, se llevaron a cabo 10 corridas para cada instancia de datos. En la *Tabla 2*, se muestran los resultados obtenidos con este algoritmo. La primera columna de la tabla muestra el número de la instancia, las columnas 2 a 4 muestran respectivamente la peor solución obtenida,

el promedio de las soluciones y la mejor solución obtenida en las 10 ejecuciones. Las columnas 5 a 7 muestran respectivamente los GAPs porcentuales de la peor solución, el promedio de las soluciones y la mejor solución obtenida en las 10 corridas con respecto a la solución óptima o a la mejor conocida. Finalmente, la octava columna muestra el tiempo promedio en segundos que le tomó al algoritmo obtener las soluciones mostradas.

**Tabla 2**

*Resultados obtenidos para el tiempo de ciclo con el algoritmo COMSOAL-2S*

Instancia	Solución (ct)			% GAP			Tiempo medio (s)
	Peor	Promedio	Mejor	Peor	Promedio	Mejor	
1	5	5	5	0.00 %	0.00 %	0.00 %	0.0304
2	7	7	7	0.00 %	0.00 %	0.00 %	0.0264
3	22	22	22	0.00 %	0.00 %	0.00 %	0.0966
4	18	18	18	0.00 %	0.00 %	0.00 %	0.1298
5	24	24	24	0.00 %	0.00 %	0.00 %	0.1285
6	35	35	35	0.00 %	0.00 %	0.00 %	0.1057
7	3	3	3	0.00 %	0.00 %	0.00 %	0.0121
8	5	5	5	0.00 %	0.00 %	0.00 %	0.0276
9	4	4	4	0.00 %	0.00 %	0.00 %	0.0293
10	638	638	638	0.00 %	0.00 %	0.00 %	0.9593
11	510	510	510	0.00 %	0.00 %	0.00 %	0.8128
12	425	425	425	0.00 %	0.00 %	0.00 %	0.6791
13	365	365	365	0.00 %	0.00 %	0.00 %	0.641
14	319	319	319	0.00 %	0.00 %	0.00 %	0.5037
15	2923	2919.4	2919	7.27 %	7.13 %	7.12 %	19.9703
16	2335	2335	2335	0.00 %	0.00 %	0.00 %	12.925
17	1946	1946	1946	0.00 %	0.00 %	0.00 %	13.6944
18	1668	1668	1668	0.00 %	0.00 %	0.00 %	12.9667
19	1460	1460	1460	0.00 %	0.00 %	0.00 %	14.6979
20	1298	1297.2	1297	0.08 %	0.02 %	0.00 %	17.4464
21	1069	1063.2	1062	0.66 %	0.11 %	0.00 %	21.4184
22	975	973.7	973	0.21 %	0.07 %	0.00 %	12.7371
Media	-	-	-	0.37 %	0.33 %	0.32 %	5.9108

A partir de los resultados obtenidos, es posible establecer ciertas conclusiones en

relación al algoritmo COMSOAL-2S propuesto. En primer lugar, como puede notarse al observar las columnas de *Peor solución* y *Mejor solución* de la *Tabla 2*, la desviación entre el mejor y peor resultado obtenidos para una misma instancia no es muy grande. De hecho, a excepción de los problemas de prueba 15, 20, 21 y 22, en la mayoría de los casos el mejor resultado, el peor resultado y el resultado promedio coinciden, lo cual es señal de la robustez del algoritmo. Es también importante notar que, a excepción de la instancia 15, el algoritmo propuesto encuentra la solución óptima o la mejor conocida en al menos una de las 10 ejecuciones.

Adicionalmente, las desviaciones porcentuales promedio son únicamente del 0.33 %, muy buenos resultados para un algoritmo de la sencillez del propuesto. Por otra parte, los tiempos de ejecución sí difieren significativamente. Aunque el modelo de solución exacta logra encontrar soluciones instantáneas en algunas de las instancias, se requirió una hora para la ejecución del modelo en otros casos, en los que ni siquiera se logró obtener la solución óptima. En contraste, el promedio de los tiempos medios de ejecución obtenidos para las 22 instancias de datos con el COMSOAL-2S es de 5.9108 segundos, y el mayor tiempo medio observado fue de solo 21.4 segundos. En general, los tiempos del método propuesto son substancialmente menores y la calidad de las soluciones iguala o incluso supera la del modelo exacto, salvo algunas excepciones.

Finalmente, al analizar los resultados del método COMSOAL-2S desarrollado, es importante considerar también las limitaciones de la presente investigación. En primer lugar, debido a las suposiciones tomadas en cuenta para la generación de los métodos descritos, el problema de balanceo de líneas bilaterales en el que se centra este estudio es una abstracción y no es un fiel reflejo de las condiciones que podrían encontrarse en

situaciones reales dentro de una línea de producción. Para simplificar el planteamiento del problema, se omiten diversos factores y restricciones que en la práctica sí podrían influir en el desempeño de una línea de ensamble. Por ejemplo, los métodos descritos consideran tiempos de tarea determinísticos, cuando en situaciones reales se presenta siempre cierto grado de variabilidad en la duración de las tareas.

En este sentido, se reconoce que el modelo propuesto es bastante sencillo. Dado que, como se estableció anteriormente, el 2S-ALBP es un problema de complejidad tipo NP-Hard, se considera que en estudios futuros este modelo COMSOAL-2S podría funcionar como una base para la generación de un modelo metaheurístico más complejo que se ajuste mejor a la naturaleza multifactorial y combinatoria del problema de balanceo de línea, permitiendo obtener resultados más cercanos a la realidad y un mejor rendimiento. Además, resultaría también de utilidad trabajar en la obtención de nuevas instancias de datos para estudios futuros. Al realizar la revisión bibliográfica, se encontró muy poca variedad en cuanto a problemas de prueba para el 2S-ALBP, sobre todo en comparación con otras variantes del problema de balanceo de líneas.

## 5. Conclusiones

La presente investigación tuvo como punto central el problema de balanceo de líneas bilaterales (2S-ALBP), una variante del problema de balanceo de líneas clásico en el que se consideran ambos lados de la línea para la asignación de tareas (Bartholdi, 1993). Desde su formulación, los autores que lo han abordado han optado por dos alternativas de solución para este problema: métodos de solución exacta y métodos heurísticos o metaheurísticos. La primera alternativa, si bien permite encontrar soluciones óptimas, puede llegar a requerir largos tiempos de cómputo. Por otro lado, la segunda alternativa, aunque no siempre permite encontrar el óptimo, sí suele generar soluciones factibles más o menos cercanas a él, requiriendo un tiempo de cómputo menor (Grzechca, 2013).

Considerando lo anterior, el objetivo principal de este estudio fue proponer un método heurístico COMSOAL-2S para la resolución del 2S-ALBP, basado en la lógica del algoritmo COMSOAL, que permitiera la obtención de soluciones factibles dentro de un tiempo de cómputo razonable. Además, se seleccionó también un modelo de la literatura, con el propósito de obtener resultados que sirvieran como punto de partida para analizar el desempeño del algoritmo heurístico propuesto. Para poder evaluar al algoritmo propuesto, se llevó a cabo una comparación entre los resultados generados por ambas alternativas de solución con 22 instancias de prueba obtenidas de la literatura.

En el caso del modelo de solución exacta, el software no pudo encontrar una solución óptima en el tiempo de ejecución límite establecido (3600 segundos) para 14 de las 22 instancias de datos utilizadas. En estos casos, el GAP porcentual promedio entre las cotas inferior y superior generadas por el software fue de 45.56 %. El GAP, aunque fue mucho más pronunciado en instancias con mayor número de tareas, puede también observarse

en instancias más pequeñas. Debido a la manera en que los modelos de solución exacta se han desempeñado en estudios similares, estos resultados no son del todo inesperados, pues es común encontrar casos en los que no pueden encontrarse soluciones óptimas para el 2S-ALBP por su complejidad.

Para el método COMSOAL-2S, dado que contiene componentes aleatorios, se llevaron a cabo 10 corridas por cada instancia de datos, reportando únicamente para cada caso la peor solución, la mejor solución y la solución promedio. Al analizar los resultados obtenidos, es posible establecer que se trata de un algoritmo robusto, pues la desviación entre la mejor y peor solución es nula en el 81.8 % de los casos, y no es muy grande en los restantes.

Además, a excepción de la instancia 15 que presentó resultados insatisfactorios, el método propuesto logró encontrar para todas las instancias la solución óptima o la mejor conocida en al menos una de las 10 ejecuciones llevadas a cabo. De hecho, las desviaciones porcentuales promedio son de tan solo 0.33 %. En cuanto a tiempos de ejecución, el COMSOAL-2S mostró también un desempeño aceptable en comparación con el modelo exacto, requiriendo tiempos substancialmente menores para generar soluciones de igual o incluso superior calidad.

Está claro que, debido a las suposiciones tomadas en cuenta para su formulación, el método propuesto desprecia ciertos factores que sí influirían en situaciones reales, como la variabilidad de los tiempos de tarea. Por ende, se reconoce que es bastante sencillo. Sin embargo, con base en los resultados obtenidos, es posible concluir que el método COMSOAL-2S presenta buenos resultados a pesar de su sencillez. Por esta razón, se considera que, en estudios futuros, podría fungir como punto de partida para la gene-

ración de un método metaheurístico más complejo que se ajuste mejor a la naturaleza multifactorial y combinatoria del 2S-ALBP.

## Referencias

- Aghajani, M., Ghodsi, R., y Javadi, B. (2014). Balancing of robotic mixed-model two-sided assembly line with robot setup times. *International Journal of Advanced Manufacturing Technology*, 74(5-8), 1005 - 1016.
- Arcus, A. (1996). Comsol a computer method of sequencing operations for assembly lines. *International Journal of Production Research*, 4(4), 259-277.
- Bartholdi, J. (1993). Balancing two-sided assembly lines: a case study. *International Journal of Production Research*, 31(10), 2447-2461. doi: 10.1080/00207549308956868
- Cerqueus, A., y Delorme, X. (2019). A branch-and-bound method for the bi-objective simple line assembly balancing problem. *International Journal of Production Research*, 57(18), 5640–5659.
- Delice, Y., Aydođan, E. K., Özcan, U., y İlkay, M. S. (2017). A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing. *Journal of Intelligent Manufacturing*, 28(1), 23–36.
- Gansterer, M., y Hartl, R. F. (2018). One- and two-sided assembly line balancing problems with real-world constraints. *International Journal of Production Research*, 56(8), 3025–3042.
- Grzechca, W. (2013). Idle times analysis in two-sided assembly line balancing problem. *IFAC Proceedings Volumes*, 46(9), 1720-1725. doi:

<https://doi.org/10.3182/20130619-3-RU-3018.00520>

- Hamzadayı, A. (2018). Balancing of mixed-model two-sided assembly lines using teaching-learning based optimization algorithm. *Pamukkale University Journal of Engineering Sciences*, 24(4), 682 - 691.
- Janardhanan, M. N., Li, Z., y Nielsen, P. (2019). Model and migrating birds optimization algorithm for two-sided assembly line worker assignment and balancing problem. *Soft Computing*, 23(21), 11263–11276.
- Kang, H.-Y., y Lee, A. H. I. (2023). An evolutionary genetic algorithm for a multi-objective two-sided assembly line balancing problem: a case study of automotive manufacturing operations. *Quality Technology & Quantitative Management*, 20(1), 66-88.
- Kim, Y. K., Kim, Y., y Kim, Y. J. (2000). Two-sided assembly line balancing: A genetic algorithm approach. *Production Planning & Control*, 11(1), 44-53.  
doi: 10.1080/095372800232478
- Kim, Y. K., Song, W. S., y Kim, J. H. (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers & Operations Research*, 36(3), 853-865.
- Kucukkoc, I., y Zhang, D. (2014). Mixed-model parallel two-sided assembly line balancing problem: A flexible agent-based ant colony optimization approach. *Computers & Industrial Engineering*, 97, 58–72.
- Kucukkoc, I., y Zhang, D. (2016). Integrating ant colony and genetic algorithms in the balancing and scheduling of complex assembly lines. *International*

*Journal of Advanced Manufacturing Technology*, 82(1-4), 265–285.

- Lee, T. O., Kim, Y., y Kim, Y. K. (2001). Two-sided assembly line balancing to maximize work relatedness and slackness. *Computers & Industrial Engineering*, 40(3), 273-292.
- Li, D., Zhang, C., Tian, G., Shao, X., y Li, Z. (2018). Multiobjective program and hybrid imperialist competitive algorithm for the mixed-model two-sided assembly lines subject to multiple constraints. *IEEE Transactions on Systems, Man & Cybernetics: Systems*, 48(1), 119–129.
- Li, Y. (2017). The type-ii assembly line rebalancing problem considering stochastic task learning. *International Journal of Production Research*, 55(24), 7334–7355.
- Li, Z., Janardhanan, M. N., Tang, Q., y Nielsen, P. (2021). Local search methods for type i mixed-model two-sided assembly line balancing problems. *Memetic Computing*, 13(1), 111-130.
- Lopes, T., Sikora, C., Michels, A., y Magatão, L. (2020). Mixed-model assembly lines balancing with given buffers and product sequence: model, formulation comparisons, and case study. *Annals of Operations Research*, 286(1/2), 475–500.
- Nkasu, M. M., y Leung, K. H. (1995). A stochastic approach to assembly line balancing. *International Journal of Production Research*, 33(4), 975.
- Özcan, U., y Toklu, B. (2009). Balancing of mixed-model two-sided assembly lines. *Computers & Industrial Engineering*, 57(1), 217–227.

- Özcan, U., y Toklu, B. (2010). Balancing two-sided assembly lines with sequence-dependent setup times. *International Journal of Production Research*, 48(18), 5363 - 5383.
- Papadopoulos, B., Yin, Q., Luo, X., y Hohenstein, J. (2021). A branch-and-price algorithm for balancing two-sided assembly lines with zoning constraints. *Journal of Mathematics*, 2021. doi: 10.1155/2021/4196228
- Pereira, J. (2018). The robust (minmax regret) assembly line worker assignment and balancing problem. *Computers and Operations Research*, 93, 27-40.
- Purnomo, H. D., Wee, H.-M., y Rau, H. (2013). Two-sided assembly lines balancing with assignment restrictions. *Mathematical and Computer Modelling*, 57(1), 189-199. (Mathematical and Computer Modelling in Power Control and Optimization) doi: <https://doi.org/10.1016/j.mcm.2011.06.010>
- Pérez O., J., Pazos R., R. A., Frausto S., J., Rodríguez O., G., Cruz R., L., y Fraire H., H. (2004). Comparison and selection of exact and heuristic algorithms. En A. Laganá, M. L. Gavrilova, V. Kumar, Y. Mun, C. J. K. Tan, y O. Gervasi (Eds.), *Computational science and its applications – iccsa 2004* (p. 415-424). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Pınarbaşı, M., Yüzükırmızı, M., y Toklu, B. (2016). Variability modelling and balancing of stochastic assembly lines. *International Journal of Production Research*, 54(19), 5761–5782.
- Saif, U., Guan, Z., Wang, B., Mirza, J., y Huang, S. (2014). A survey on assembly lines and its types. *Frontiers of Mechanical Engineering*, 9(2), 95-105. doi:

<https://doi.org/10.1007/s11465-014-0302-1>

- Sivasankaran, P., y Shahabudeen, P. (2014). Literature review of assembly line balancing problems. *Int J Adv Manuf Technol*, 73, 1665–1694.
- Taha, R. B., El-Kharbotly, A., Afia, N., y Sadek, Y. (2012). Mixed-model two-sided assembly line balancing using a modified genetic algorithm. *Proceedings of the International Conference on Industrial Engineering & Operations Management*, 1168 - 1177.
- Türkmen, A., Yesil, Y., y Kayar, M. (2016). Heuristic production line balancing problem solution with matlab software programming. *International Journal of Clothing Science and Technology*, 28(6), 750-779.
- Wilson, J. (2014). Henry ford vs. assembly line balancing. *International Journal of Production Research*, 52(3), 757–765.
- Yadav, A., Verma, P., y Agrawal, S. (2020). Mixed model two-sided assembly line balancing problem: an exact solution approach. *International Journal of Systems Assurance Engineering & Management*, 11(2), 335–348.
- Yang, C., Gao, J., y Jinlin, L. (2014). Balancing mixed-model assembly lines with adjacent task duplication. *International Journal of Production Research*, 52(24), 7454–7471.
- Yang, Z., Zhang, G., y Zhu, H. (2016, 08). Multi-neighborhood based path relinking for two-sided assembly line balancing problem. *Journal of Combinatorial Optimization*, 32. doi: 10.1007/s10878-015-9959-6
- Yin, Q., Luo, X., Sun, J., y Zhang, L. (2021). A workstation solution based heu-

ristic algorithm for assembly line balancing problem. *2021 33rd Chinese Control and Decision Conference (CCDC), Control and Decision Conference (CCDC), 2021 33rd Chinese*, 4010-4015.

Yuwei, L., Jian'e, C., Feng, Z., y Shijing, Z. (2020). A research of multi-constrained two-sided mixed-model assembly line balancing problem based on genetic algorithm. *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, 377-381.

## A. Apéndice A

En esta sección, se muestran las instancias de datos empleadas como problemas de prueba en la presente investigación. Se utilizaron una instancia de 9 tareas, cuatro de 12 tareas (Kim y cols., 2000), una de 16 tareas (Papadopoulos y cols., 2021), tres de 24 tareas, cinco de 65 tareas (Z. Yang y cols., 2016) y ocho de 205 tareas (Lee y cols., 2001).

**Tabla 3**

*Instancias de datos 1 y 2*

Instancia 1: 3 estaciones acopladas Instancia 2: 2 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	2	L	-
2	3	R	-
3	2	E	-
4	3	L	1
5	1	E	2
6	1	L	3
7	3	E	4, 5
8	3	R	5
9	2	E	5, 6
10	2	E	7, 8
11	2	E	9
12	1	R	11

**Tabla 4***Instancia de datos 3*

16 tareas, 2 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	6	E	-
2	5	E	-
3	2	L	1
4	9	E	1
5	8	R	2
6	4	L	3
7	7	E	4, 5
8	4	E	6, 7
9	5	R	7
10	4	R	7
11	6	E	8
12	5	L	9
13	6	E	9, 10
14	4	E	11
15	3	E	11, 12
16	4	E	13

**Tabla 5***Instancias de datos 4, 5 y 6*

Instancia 4: 4 estaciones acopladas Instancia 5: 3 estaciones acopladas Instancia 6: 2 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	3	L	-
2	7	L	-
3	7	R	-
4	5	R	-
5	4	L	2
6	3	E	2, 3
7	4	R	3
8	3	E	5
9	6	E	6
10	4	E	7
11	4	L	1
12	3	L	8, 9
13	3	E	9
14	9	R	9, 10
15	5	R	4
16	9	L	11
17	2	E	12
18	7	E	13
19	9	E	13, 14
20	9	R	15
21	8	L	16, 17
22	8	E	18
23	9	R	19, 20
24	9	E	20

**Tabla 6***Instancia de datos 7*

3 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	2	L	-
2	3	R	-
3	2	E	-
4	3	L	1
5	1	R	2
6	1	E	2, 3
7	2	E	4, 5
8	2	L	5
9	1	E	6

**Tabla 7***Instancias de datos 8 y 9*

Instancia 8: 3 estaciones acopladas Instancia 9: 4 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	2	L	-
2	3	R	-
3	2	E	-
4	3	L	1
5	1	E	2
6	1	L	3
7	3	E	4, 5
8	3	R	5
9	2	E	5, 6
10	2	E	7, 8
11	2	E	9
12	1	R	11

**Tabla 8***Instancias de datos 10-14*

Instancia 10: 4 estaciones acopladas Instancia 11: 5 estaciones acopladas Instancia 12: 6 estaciones acopladas Instancia 13: 7 estaciones acopladas Instancia 14: 8 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	49	E	-
2	49	E	-
3	71	E	1, 2
4	26	E	3
5	42	E	4
6	30	E	4
7	167	R	4
8	91	R	7
9	52	L	4
10	153	L	9
11	68	E	4
12	52	E	4
13	135	E	-
14	54	E	5, 6, 8, 10, 11, 12, 13
15	57	E	14
16	151	L	15
17	39	L	16
18	194	R	14
19	35	R	18
20	119	E	14
21	34	E	19, 20
22	38	E	14
23	104	E	3
24	84	E	23
25	113	L	4
26	72	R	4
27	62	R	4
28	272	R	27
29	89	L	-
30	49	L	-

**Tabla 9***Instancias de datos 10-14, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
31	11	E	17, 21, 22, 24, 25, 26
32	45	E	31
33	54	E	32
34	106	E	33
35	132	R	34
36	52	E	31
37	157	E	36
38	109	E	37
39	32	L	38
40	32	R	38
41	52	E	4
42	193	E	41
43	34	E	42
44	34	R	-
45	97	L	4
46	37	E	44, 45
47	25	L	46
48	89	L	47
49	27	E	4
50	50	E	28, 29, 30, 35, 39, 40, 48, 49
51	46	R	31
52	46	E	31
53	55	L	31
54	118	E	31
55	47	R	31
56	164	E	31
57	113	E	56
58	69	L	31
59	30	R	31
60	25	E	31
61	106	R	31
62	23	E	31, 43
63	118	L	62
64	155	L	63
65	65	E	50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 64

**Tabla 10***Instancias de datos 15-22*

Instancia 15: 4 estaciones acopladas Instancia 16: 5 estaciones acopladas Instancia 17: 6 estaciones acopladas Instancia 18: 7 estaciones acopladas Instancia 19: 8 estaciones acopladas Instancia 20: 9 estaciones acopladas Instancia 21: 11 estaciones acopladas Instancia 22: 12 estaciones acopladas			
Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
1	692	-	E
2	42	-	E
3	261	2	R
4	261	2	L
5	157	3, 4	E
6	90	-	E
7	54	5	R
8	67	7	R
9	30	8	R
10	106	9	R
11	32	10	R
12	62	11	R
13	54	5	L
14	67	13	L
15	30	14	L
16	106	15	L
17	32	16	L
18	62	17	L
19	56	-	E
20	67	-	E
21	86	-	E
22	37	20, 21	E
23	41	22	E
24	72	23	E
25	86	-	R
26	16	24	L

**Tabla 11***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
27	51	24	R
28	66	24, 25	R
29	41	28	R
30	72	29	R
31	51	30	R
32	16	30	R
33	15	29	R
34	15	23	L
35	85	26, 27, 31, 32, 33, 34	E
36	59	1, 6, 12, 18, 19	E
37	23	36	L
38	13	37	L
39	19	38	L
40	108	36	E
41	214	36	E
42	80	36	E
43	37	40, 42	L
44	84	43	L
45	18	39, 44	L
46	12	45	L
47	29	46	L
48	37	45	L
49	13	48	L
50	70	49	L
51	217	45	L
52	72	51	L
53	85	45	L
54	43	40, 42	R
55	97	54	R
56	37	55	R
57	13	56	R
58	35	57	R
59	217	55	R
60	72	59	R
61	85	55	R
62	25	36	E
63	37	62	E
64	37	63	E

**Tabla 12***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
65	103	64	E
66	140	65	E
67	49	66	E
68	35	64	E
69	51	36	E
70	88	69	E
71	53	70	E
72	144	36	E
73	337	71, 72	E
74	107	73	E
75	371	36	E
76	97	74	E
77	166	76	E
78	92	76	L
79	92	76	R
80	106	67, 68, 77, 78, 79	E
81	49	80	E
82	92	77	E
83	371	36	E
84	87	81	E
85	162	84	E
86	96	85	E
87	79	86	E
88	96	85	E
89	42	88	E
90	88	85	R
91	90	90	R
92	97	41, 47, 50, 53, 58, 60, 61, 75, 82, 83, 87, 89, 91	R
93	270	92	R
94	452	92	E
95	48	92	R
96	338	92	E

**Tabla 13***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
97	34	92	E
98	65	92	E
99	50	92	E
100	112	97, 98, 99	E
101	48	100	E
102	117	101	E
103	50	100	E
104	68	103	R
105	232	100	L
106	122	105	L
107	151	105	E
108	31	106, 107	L
109	97	100	E
110	308	36	R
111	116	36	L
112	312	36	R
113	34	95, 96, 102, 104, 108, 109, 110, 111, 112	E
114	128	113	L
115	54	113	E
116	175	113	R
117	55	113	E
118	306	113	E
119	59	113	E
120	59	113	E
121	66	113	E
122	66	113	E
123	23	113	E
124	244	113	E
125	54	-	E
126	294	118, 119, 120, 121, 122, 123, 125	R
127	84	126	E
128	61	126	E

**Tabla 14***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
129	57	126	E
130	38	100	R
131	944	100	E
132	511	131	R
133	625	132	R
134	445	100	R
135	68	93, 94, 128, 129	L
136	53	130, 135	L
137	49	135	E
138	92	135	E
139	236	135	E
140	116	135	L
141	265	135	L
142	149	135	L
143	74	140, 141, 142	L
144	332	135	E
145	324	135	E
146	104	145	L
147	51	135	L
148	58	135	R
149	67	135	R
150	49	135	R
151	107	135	E
152	38	135	L
153	27	135	L
154	68	153	E
155	207	154	E
156	202	155	E
157	83	156	E
158	35	135	R
159	58	158	R
160	42	114, 115, 116, 117, 137, 138, 139, 143, 144, 146, 147, 148, 149, 150, 151, 152	E

**Tabla 15***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
161	68	113	R
162	68	113	R
163	68	113	R
164	103	163	R
165	103	162, 164	R
166	103	165	R
167	103	161, 166	R
168	103	167	R
169	68	113	L
170	103	160	L
171	68	169, 113	L
172	103	170, 171	L
173	103	172	L
174	68	113	L
175	103	173, 174	L
176	103	175	L
177	10	168, 176	E
178	187	160	E
179	134	160	L
180	89	178, 179	L
181	58	180	L
182	49	181	L
183	134	180	L
184	53	160	L
185	334	177	E
186	24	177	R
187	76	177	R
188	76	177	L
189	192	133, 134, 136, 157, 159, 185, 186, 187, 188	E
190	98	189	E
191	258	189	R
192	165	191	E

**Tabla 16***Instancias de datos 15-22, continuación*

Tarea No.	Tiempo de tarea	Lado de operación	Predecesoras
193	38	189	R
194	115	177	E
195	83	177	L
196	56	195	R
197	29	194, 196	R
198	303	197	R
199	18	197	R
200	29	199	R
201	154	197	L
202	90	201	L
203	93	113	E
204	94	113	E
205	165	113	E