

Heurística de enfoque bidireccional para la solución de problemas de equilibrado de líneas de ensamble

Heuristic of Bidirectional Approach to Solve Assembly Line Balancing Problems

Jorge Michael Burgos Meneses ⁽¹⁾

⁽¹⁾ Ingeniero Industrial, Universidad de Córdoba, Montería-Córdoba, Colombia. ing.jmburgos.09@gmail.com

Recibido 15 de agosto de 2014. Modificado 12 de diciembre de 2014. Aprobado 14 de enero de 2015.

DOI: <http://dx.doi.org/10.16924/riua.v0i41.667>

Palabras clave

Algoritmos, equilibrado, heurística, líneas de ensamble, tiempo de ciclo.

Key words

Algorithm, assembly line, balance, heuristic, cycle time.

Resumen

En este artículo se describe y evalúa un algoritmo de enfoque bidireccional para resolver los SALBPs-1, el cual se fundamenta en la heurística JOMI, construyendo simultáneamente soluciones desde ambos lados de la matriz de precedencias. Aquí se muestra una evaluación comparativa entre este procedimiento y otros 14 algoritmos similares, utilizando 269 instancias encontradas en la literatura, donde los resultados evidenciaron que el algoritmo propuesto genera muy buenas soluciones al arrojar el mejor promedio de eficiencia con un 91,42%. De este modo, se presenta otra forma sencilla y factible de resolver los problemas de líneas de ensamble.

Abstract

This article describes and evaluates a bidirectional approach algorithm to solve SALBPs-1. Such a procedure is based on heuristic JOMI, by building solutions from both sides of the precedence matrix simultaneously. A comparative evaluation among this procedure and 14 other similar algorithms is shown. Two hundred and sixty-nine instances found in the literature were used, the results illustrate that the proposed algorithm produces very good solutions, in this case, throwing the best average efficiency of 91.42%. Thus, another simple and practical way to solve assembly lines problems is presented.

INTRODUCCIÓN

Las líneas de ensamble son componentes esenciales para numerosos procesos industriales de producción, principalmente los asociados a la fabricación de productos similares, en masa y baja variedad (celulares, computadores, televisores, carros, etc). Estas líneas se distinguen por estar formadas por estaciones de trabajo, a través de las cuales circula el producto en proceso hasta ser terminado. Cada una de estas estaciones cuenta con un tiempo idéntico para realizar las operaciones, tareas o actividades asignadas, llamado “*tiempo de ciclo*”.

Por otro lado, el “*equilibrado de líneas de ensamble*” consiste en la adecuada asignación de tareas a estaciones de trabajo, de tal forma que se respeten las relaciones de precedencia que pueden presentar algunas tareas entre sí, y el tiempo de capacidad de las estaciones de trabajo. Este problema de asignación de tareas es conocido como problema de balanceo de líneas de ensamble (*Assembly Line*

Balancy Problem, ALBPs) y ha sido clasificado en dos tipos principales, los problemas simples de balanceo de líneas de ensamble (*Simple Assembly Line Balancy Problem, SALBP*) y los problemas generales de balanceo de líneas de ensamble (*General Assembly Line Balancy Problem, GALBP*); esta clasificación se detalla en (Baybars, 1986), donde se estudia el desarrollo de los SALBP a través del tiempo.

Los SALBP están limitados sólo por restricciones de precedencia y capacidad (*tiempo*); sus instancias se pueden representar de la siguiente manera: un grupo de actividades i ($i=1,\dots,n$), las cuales son indivisibles, poseen una duración de procesamiento conocida t_i y algunas presentan relaciones de precedencia entre sí; se distribuyen en un grupo de estaciones de trabajo S_j ($j=1,\dots,m$), las cuales presentan un tiempo de duración idéntico (*tiempo de ciclo* “ C ”), con el fin de maximizar la eficiencia de la línea de ensamble, ya sea minimizando el número de estaciones de trabajo, el

tiempo de ciclo o ambos parámetros simultáneamente dependiendo del tipo de problema. En el amplio estado de arte (Scholl & Becker, 2006) se detallan las investigaciones más sobresalientes referentes a estos problemas, así como la clasificación de estos según la función objetivo en SALBP-1 (cuando se busca minimizar el número de estaciones dado el tiempo de ciclo), SALBP-2 (cuando se busca minimizar el tiempo de ciclo dado el número de estaciones), SALBP-E (cuando se busca maximizar la eficiencia variando tanto el tiempo de ciclo como el número de estaciones) y SALBP-F (cuando se busca encontrar una factibilidad dados estos dos parámetros).

Los GALBP son problemas más generales que consideran otras restricciones aparte del tiempo de ciclo y las relaciones de precedencia, como el caso de estaciones en paralelo, donde en Becker & Scholl (2009) se define y modela este tipo de problema, además de proponer un procedimiento para su solución basado en “branch and bound”; líneas en forma de U donde en Sabuncuoglu, Erel & Alp (2009) este tipo de problema se introduce y se utiliza un algoritmo de colonias de hormigas para resolverlo; incompatibilidad entre tareas, en Tuncel & Topaloglu (2013) se presenta un problema de balanceo de líneas de ensamble de una empresa de fabricación electrónica que trae consigo esta restricción junto con estaciones en paralelo. A pesar de que los GALBP se caracterizan por ser problemas más generales y reales, la mayoría de investigadores han dedicado sus estudios a los SALBP. Esto se debe a que sus soluciones pueden ser adaptadas a las soluciones de los GALBP; en Becker & Scholl (2006) se realiza un estudio completo de las investigaciones relacionadas con este tipo de problemas.

En lo referente a métodos de solución para los ALBP, se observa que existen numerosos algoritmos para su resolución, como se evidencia en el estado del arte (Scholl & Becker, 2006) y el proyecto de clasificación de balanceo de líneas de ensamble (Boysen, Fließner & Scholl, 2007). Entre estos métodos se destacan: *Los métodos heurísticos*, los cuales generalmente utilizan reglas de prioridad para su funcionamiento y generan buenas respuestas en tiempos computacionales muy cortos, estos métodos a pesar de no asegurar la solución óptima del problema son frecuentemente usados y esto se debe a su fácil manejo, aplicación e implementación, en Talbot, Patterson & Gehrlein (1986) se observa un experimento computacional para evaluar la eficiencia de 26 heurísticas de decisión y en Burgos, Burgos, Garzón & López (2013) se observa el diseño y evaluación de una nueva heurística greedy; *los métodos basados en procedimientos de búsqueda exacta “branch and bound”*, los cuales se caracterizan por generar las respuestas más eficientes, en su mayoría óptimos aunque en tiempos de procesamiento más largos, entre estos se destacan los algoritmos FABLE (Johnson, 1988) y SALOME (Scholl & Klein, 1997); y finalmente se tienen *los métodos basados en la implementación de metaheurísticas*, los cuales son

procedimientos que no garantizan la solución óptima de un problema, sin embargo, se diferencian de las heurísticas en que orientan su búsqueda evitando óptimos locales, entre estos se destacan las siguientes investigaciones: la de Rubinovitz & Levitin (1995), que describe un algoritmo genético para la creación de múltiples soluciones en los ALBP, el trabajo de Lapierre, Ruiz & Soriano (2006), en el cual se presenta un algoritmo de búsqueda tabú y se observan sus diferencias con otros procedimientos, y el estudio de Bautista & Pereira (2007) en el que se aplica el algoritmo de colonia de hormigas en un problema de equilibrado de líneas de ensamble.

En esta investigación se presenta el diseño, evaluación, comparación y estudio del comportamiento, en un caso real, de un algoritmo de enfoque bidireccional basado en la heurística JOMI. Esto con el fin de ofrecer un método alternativo para la solución de los SALBP-1.

HEURÍSTICAS GREEDY

En lo referente a la resolución exacta de los problemas de balanceo de líneas de ensamble se observa que existen numerosos y notables avances, prueba de ello son los algoritmos FABLE (Johnson, 1988) y SALOME (Scholl & Klein, 1997); no obstante, las heurísticas siguen ocupando un lugar importante en este asunto debido a que generan soluciones adecuadas en tiempos de procesamiento relativamente cortos, son fáciles de manejar y son buenas alternativas para los problemas en que los métodos exactos no tienen éxito. En Silver (2004) se muestran las ventajas que trae consigo la implementación de este tipo de procedimientos.

Las heurísticas greedy utilizan datos disponibles del problema para construir paso a paso una solución del mismo. Estos algoritmos son utilizados constantemente por los investigadores para el desarrollo de nuevos procedimientos, como se observa en el trabajo de Feo & Resenden (1995) que define diversos componentes comprendidos en *Greedy Randomized Adaptive Search Procedures* (GRASP) y donde se demuestra cómo desarrollar heurísticas para problemas de optimización combinatoria; la investigación de Bautista, Fernández, González y Laguna (2005) expone un procedimiento de combinación de soluciones representadas por secuencias de reglas de prioridad de heurísticas greedy bajo un esquema *scatter search*; y el estudio de Chica, Cerdón, Damas y Bautista (2010) muestra dos nuevas propuestas multi-objetivo basadas en optimización de colonias de hormigas y heurísticas greedy de búsqueda aleatoria para resolver extensiones más realistas del problema.

La Figura 1 muestra un esquema general del funcionamiento de las heurísticas greedy aplicadas en los SALBP-1, donde se describe paso a paso dicho procedimiento.

Cabe destacar, teniendo en cuenta el esquema de trabajo mostrado en la Figura 1, que si después de eliminar todas las referencias de una actividad y calcular el nuevo tiempo

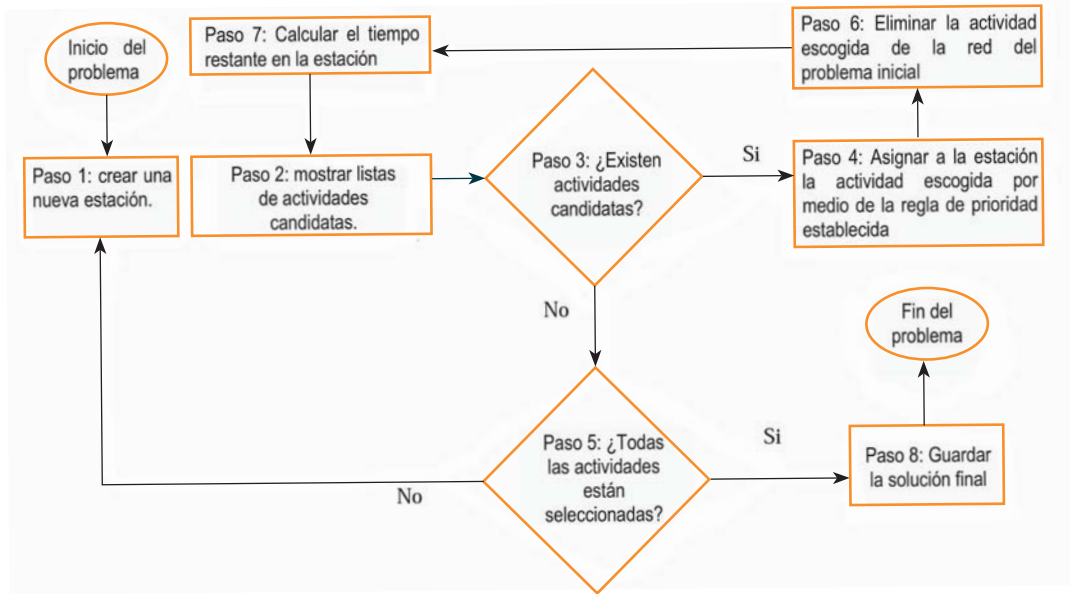


Figura 1. Esquema de un procedimiento heurístico tipo greedy para la solución de los SALBP-1

disponible en la estación, las actividades o sucesiones de actividades existentes no cumplen con las restricciones de tiempo, se crea una nueva estación con *tiempo disponible igual* a *C* y se continúa el problema con este. Por otro lado, si después de eliminar todas las referencias de una actividad y calcular el nuevo *tiempo disponible en la estación* no existen más actividades, el problema ha sido resuelto y todas las actividades han sido asignadas a las estaciones de trabajo.

El Paso 4 muestra que para asignar las tareas a las estaciones de trabajo, se debe establecer una regla de prioridad, la cual define a la heurística en cuestión. En la Tabla 1, se indican las heurísticas más utilizadas y conocidas en la literatura, además de la abreviatura que se utilizará para ellas en el presente trabajo.

HEURÍSTICA JOMI

La heurística tipo greedy llamada “JOMI” es un algoritmo presentado en Burgos et al. (2013) que se fundamenta en construir estaciones de trabajo utilizando como regla de prioridad, “escoger la primera actividad de la sucesión de actividades que genera menor tiempo ocioso en la estación”, y define a las sucesiones de actividades como un conjunto de actividades que se preceden entre sí, formando una especie de cadena y presentando un tiempo de ejecución menor o igual que el tiempo disponible de la estación. En Burgos et al. (2013) se presenta el diseño, descripción, comparación y evaluación de este algoritmo heurístico, además de su comportamiento en un caso real; en los resultados computacionales de las pruebas realizadas en dicha investigación, se observa que para una muestra de 269 instancias, de los 15 algoritmos mostrados en la Tabla 1, la heurística JOMI arroja el mejor promedio de eficiencia con un 90,2409% y, adicionalmente, en 149 de los 269 casos obtuvo soluciones óptimas.

Los buenos resultados del algoritmo JOMI en la investigación (Burgos et al., 2013) impulsaron el interés de ampliar la aplicabilidad del mismo, por lo que se procedió a diseñar otro método de solución de los SALBP-1 fundamentado en esta heurística, “enfoque bidireccional de la heurística JOMI”.

Heurísticas tipo greedy	
Nombre	Abreviatura
Mayor tiempo de operación	TOL
Mayor número de sucesoras inmediatas	MIS
Mayor número de sucesoras	MS
Mayor peso posicional	PP
Mayor peso medio posicional	PMP
Menor cota superior	MCS
Menor : (cota superior/sucesoras)	MCS/S
Mayor: (tiempo de operación/cota superior)	TOL/CS
Menor cota inferior	MCI
Menor holgura	MH
Mayor: (Sucesoras/holgura)	S/H
Bhattacharjee & Sahu	B&S
Kilbridge & Wester	K&W
Menor tiempo de operación	TOC
Heurística JOMI	JOMI

Tabla 1. Definición de las heurísticas tipo greedy

ENFOQUE BIDIRECCIONAL

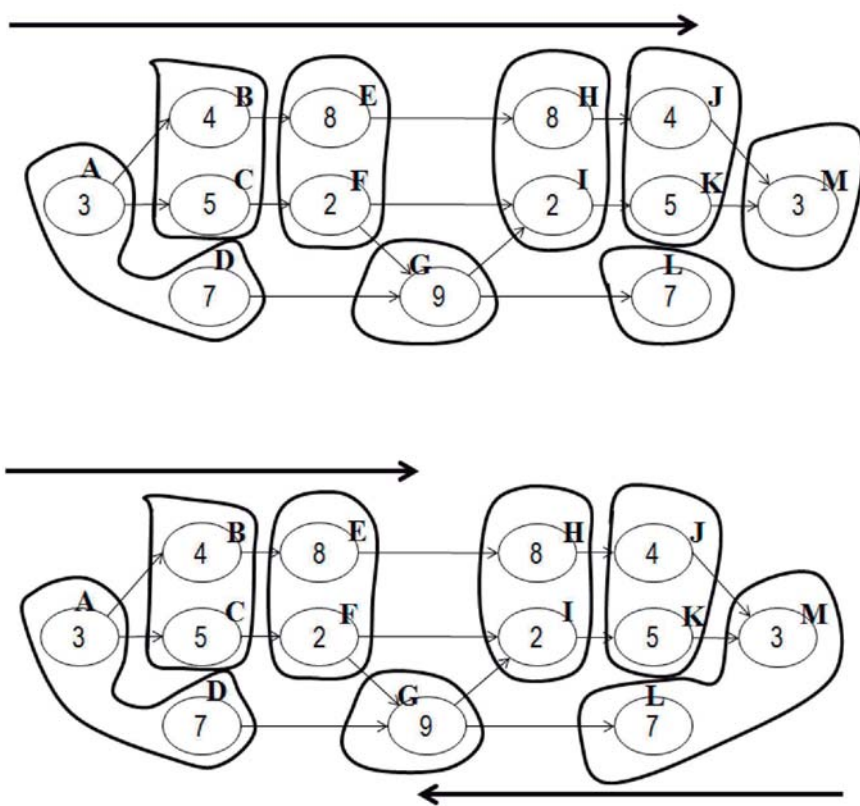
Las heurísticas greedy generan buenos resultados a un costo computacional razonable, sin embargo, tienen diversas deficiencias como es el caso que se da al finalizar el equilibrado de líneas, donde se observa una tendencia en las últimas estaciones de presentar mayor tiempo ocioso; del mismo modo, si estos algoritmos inician en dirección opuesta, de atrás hacia adelante, el tiempo de inactividad a la izquierda en una o varias de las estaciones de trabajo tendrá la misma tendencia.

Para superar esta deficiencia, se ha decidido implementar un enfoque bidireccional a la heurística JOMI y a otras de tipo greedy, con el cual se procederá a buscar soluciones generando estaciones de trabajo desde ambos lados de la matriz de precedencia, desde direcciones opuestas de forma simultánea. Luego, se escogerá de las estaciones generadas la que tenga menor tiempo ocioso, se eliminarán todas las referencias de las actividades que se encuentran en dicha estación, y se repetirán los pasos anteriores hasta que todas las actividades hayan sido seleccionadas. Este mecanismo ha sido propuesto por Scholl & Voss (1996), y utilizado por Scholl & Klein (1997) y Krzysztof & Khalil (2003) para el diseño de los algoritmos heurísticos como SALOME y Multi-Hoffmann. Los resultados de estas investigaciones

muestran las notables ventajas que traen consigo este tipo de algoritmos, ya que de cierta forma duplican el rango de búsqueda de estaciones, generan menor tiempo ocioso en la asignación de actividades y, por tanto, generan soluciones más eficientes.

El ejemplo de la Figura 2 muestra los resultados del balanceo de una línea de ensamble por medio de la heurística JOMI y su enfoque bidireccional. Aquí los nodos y los números que se encuentran en su interior, representan las tareas y sus respectivas duraciones, mientras que las relaciones de precedencia son representadas por medio de grafos. Como se observa en el primer balanceo, “*el conseguido por medio de la heurística JOMI*”, se obtienen 8 estaciones (las últimas de estas con mayor tiempo ocioso) generando una eficiencia total de 83,75%; por otro lado, en el segundo balanceo, “*el conseguido por medio del enfoque bidireccional de la heurística JOMI*”, se obtienen 7 estaciones generando una eficiencia total de 95,71%. Una de las razones fundamentales por las cuales se consigue dicho aumento en la eficiencia es que en el último caso se amplía el número de opciones en la búsqueda de una estación más eficiente.

Nuestro interés radica en el algoritmo de enfoque bidireccional de la heurística JOMI, por tanto, se realizará



Heurística JOMI

Número de estaciones	Tiempo estación	Tiempo ocioso
1	10	0
2	9	1
3	10	0
4	9	1
5	10	0
6	7	3
7	9	1
8	3	7

Enfoque bidireccional de la Heurística JOMI

Número de estaciones	Tiempo estación	Tiempo ocioso
1	10	0
2	9	1
3	10	0
4	9	1
5	10	0
6	9	1
7	10	0

Figura 2. Ejemplo balanceo de líneas de ensamble.

una evaluación comparativa entre el enfoque bidireccional de todas las heurísticas expuestas en la Tabla 1, y luego se identificarán las ventajas comparativas que trae consigo este procedimiento.

EXPERIMENTO

En las investigaciones llevadas a cabo por Barr, Golden, Kelly, Resende & Stewart (1995), Hooker (1995) y Rardin & Uzsoy (2001), se llegó a la conclusión de que no existe un procedimiento de comparación de algoritmos heurísticos que tenga una aceptación general, por tanto, para realizar las pruebas computacionales de este estudio se optó por efectuar la práctica más común, comparando las heurísticas en cuestión en un grupo relativamente grande de instancias.

PRUEBA COMPUTACIONAL

Para el desarrollo del experimento se comparó el enfoque bidireccional de todas las heurísticas mostradas en la Tabla 1. Las pruebas computacionales se realizaron en el software MATLAB 2009b, utilizando un computador con procesador de 2.10 Ghz. Intel(R) Core(TM) i3-2310M y 8GB de memoria RAM, y las instancias empleadas se encuentran en el “*benchmark data set*” para SALBP-1 del capítulo 7.1 de Scholl (1999).

Cabe aclarar que se utilizaron 269 instancias para realizar las pruebas computacionales. Estas instancias son frecuentemente usadas en investigaciones y estudios referentes al desarrollo de algoritmos heurísticos para la solución de los SALBP-1; prueba de ello es la evidencia en trabajos como Kilinc (2011) que expone el desarrollo de un nuevo algoritmo basado en aproximación de la red de Preti para solucionar ALBP, y Yeh & Kao (2009) que presenta una heurística basada en aproximación bidireccional y el método de ruta crítica, entre otros estudios.

RESULTADOS

Para tener una mejor apreciación de los resultados de las pruebas computacionales, se ha optado por dividir las instancias en tres grupos teniendo en cuenta el número de actividades de cada instancia, de esta forma se tendrían instancias pequeñas (con menos de 33 actividades), instancias medianas (con más de 33 actividades pero menos de 100 actividades) e instancias grandes (con más de 100 actividades). En la Tabla 2 donde “*n*” es el número de actividades de cada instancia y los algoritmos están dados por la abreviatura mostrada en la Tabla 1 precedida por la palabra BIDI, se muestran los promedios de eficiencia de cada algoritmo en cada grupo de instancias, además del promedio de eficiencia en todas las instancias y el tiempo de procesamiento promedio para cada algoritmo.

Al verificar cuál de los algoritmos genera mejor eficiencia, se encontró que el enfoque bidireccional de la heurística

JOMI y la misma heurística JOMI arrojan mejores resultados que los demás algoritmos puestos a prueba; el enfoque bidireccional fue superior en todos los grupos de instancias con un promedio de eficiencia total de 91,4292%, seguido de la heurística “*JOMI*” con 90,2409% y “*BIDI:B&S*” con 90,1593%. Por otro lado, en cuanto a lo relacionado con el tiempo de procesamiento, se observa que la heurística que más tarda en procesar una solución es el enfoque bidireccional de la heurística JOMI con una duración promedio de 0.5269 segundos, los demás algoritmos puestos a prueba tienen un promedio de duración similar, el cual oscila entre 0,16 y 0,3 segundos.

Ahora al enfocarnos en el procedimiento heurístico JOMI y su enfoque bidireccional, algoritmos que generaron las mejores eficiencias en las pruebas realizadas, tenemos que en 67 de los 269 casos dicho enfoque bidireccional generó mejores eficiencias que la heurística JOMI; en 201 casos ambos algoritmos generaron la misma eficiencia y en un solo caso la heurística JOMI generó mejor eficiencia que su enfoque bidireccional. A pesar de ello, al utilizar el software STATGRAPHICS centurión y considerar un nivel del 95,0% de confianza, se observó que el enfoque bidireccional de la heurística JOMI mantiene diferencia significativa con todos los algoritmos testeados a excepción de la misma heurística JOMI, ya que, para que exista diferencia significativa, la diferencia entre los promedios de eficiencia tiene que ser superior a 1,254%, y para el caso de la heurística JOMI y su enfoque bidireccional la diferencia es de 1,188%, una cifra cercana pero no superior a la establecida.

Al comparar los mejores resultados encontrados en la literatura (en su mayoría soluciones óptimas) con los resultados arrojados por el enfoque bidireccional de la heurística JOMI se nota que este algoritmo obtiene en el 69,52% de los casos (187 de las 269 instancias) la solución óptima. En la Tabla 3, se mostrará el número de soluciones óptimas conseguidas, el porcentaje de este, la desviación promedio de los resultados y la máxima desviación (máxima diferencia entre el número de estaciones óptimo y el número de estaciones generados por el algoritmo propuesto en una instancia).

APLICACIÓN

A continuación, se mostrará un breve resumen de cómo sería el comportamiento del enfoque bidireccional de la heurística JOMI en un caso real, tomando como ejemplo el caso de una fábrica de mediana escala ubicada en MIDC Amravati, “*Jadhao Gears PVT LTD*”, la cual se encarga de la producción de máquinas desmotadoras.

En la investigación realizada por Pachghare & Dalu (2012) se muestran detalladamente los tiempos de las actividades y las relaciones de precedencia de una línea de ensamble de 26 actividades de la empresa *Jadhao Gears PVT LTD*. En dicho trabajo se realizó el balanceo de esta línea con las heurísticas TOL, K&W y PP, utilizando como tiempos de ciclo 12 y 10 minutos, los cuales hacen referencia al tiempo de ciclo de

Algoritmo	Tamaño de instancias			Eficiencia promedio para todas las instancias	Tiempo de procesamiento promedio para todas las instancias (s.)
	Eficiencia promedio para instancias pequeñas (n< 33)	Eficiencia promedio para instancias medianas (33<n<100)	Eficiencia promedio para instancias grandes (100 <n <300)		
BIDI:TOC	81,2711%	81,3397%	85,8445%	82,6304%	0,16025828
BIDI:MCS	82,9426%	84,4073%	90,1473%	85,7396%	0,25237108
BIDI:MCI	82,9426%	84,4315%	90,1473%	85,7512%	0,25863746
BIDI:MCS/S	83,3610%	84,5573%	90,3239%	85,9581%	0,26009748
BIDI:K&W	83,4413%	86,3028%	92,0537%	87,3214%	0,26396482
BIDI:MIS	84,5891%	85,3520%	92,9680%	87,3874%	0,18298648
BIDI:PMP	86,1268%	86,7235%	94,9298%	88,9677%	0,27675029
BIDI:TOL	86,5117%	87,6554%	94,2415%	89,3058%	0,17804815
BIDI:MS	85,7718%	87,6824%	95,1461%	89,4133%	0,27338381
BIDI:TOL/CS	86,7293%	87,8156%	94,4723%	89,4995%	0,26942157
BIDI:S/H	86,6515%	87,8810%	95,1133%	89,6993%	0,31537957
BIDI:PP	86,7512%	88,1408%	95,4580%	89,9474%	0,29925091
BIDI:MH	86,9008%	88,1675%	95,4878%	90,0029%	0,30123733
BIDI:B&S	87,3539%	88,4058%	95,2755%	90,1593%	0,27737209
JOMI	87,2150%	88,5287%	95,4608%	90,2409%	0,25562717
BIDI:JOMI	88,3636%	89,4643%	97,1015%	91,4292%	0,52699168

Tabla 2. Promedios de eficiencia

Resultados del enfoque bidireccional de la heurística JOMI				
Tamaño de instancias (n= número de actividades)	Pequeñas (n< 33)	Medianas (33<n<100)	Grandes (100 <n <300)	Todas las instancias
Total instancias	61	130	78	269
Promedio de eficiencia	88,3636%	89,4643%	97,1015%	91,4292%
N° Soluciones óptimas	58	96	33	187
N° Soluciones óptimas	95,08 %	73,84 %	42,31 %	69,52 %
Desviación promedio	0,04918	0,3	0,5897	0,3261
Máxima desviación del óptimo	1	2	2	2

Tabla 3. Resultados de las pruebas computacionales

la producción actual y el necesario para un incremento en la producción respectivamente. La tabla 4 muestra los resultados del balanceo de dicha línea por medio de la heurística JOMI y su enfoque bidireccional, utilizando los tiempos de ciclo mostrados en Pachghare & Dalu (2012).

Como se puede observar, los resultados arrojados por los algoritmos heurísticos son muy buenos ya que tanto la heurística JOMI como su enfoque bidireccional obtuvieron el número teórico mínimo de estaciones para el caso de la producción actual (tiempo de ciclo 12 minutos), lo cual supera en 14% el balanceo de la línea mostrado en dicha investigación. Por otro lado, para el caso del tiempo de ciclo

10 minutos vemos que a pesar de no conseguir el número teórico mínimo de estaciones se ha conseguido una eficiencia superior al 80%.

CONCLUSIONES

La presente investigación propone un algoritmo de enfoque bidireccional para la solución de los SALBP-1. Este algoritmo se fundamenta en la heurística JOMI, asignando las actividades a las estaciones de trabajo desde ambos lados de la matriz de precedencia simultáneamente.

Generalidades		(Vrittika & Dalu, 2012)		JOMI		ENFOQUE BIDIRECCIONAL	
Tiempo de ciclo	N° teórico mínimo de estaciones	Estaciones	Eficiencia	Estaciones	Eficiencia	Estaciones	Eficiencia
12	5	6	68,5%	5	82,31%	5	82,31%
10	5	6	82,31%	6	82,31%	6	82,31%

Tabla 4. Resultados de la aplicación de la heurística JOMI y su enfoque bidireccional en el caso de la empresa Jadhao Gears PVT LTD

En los resultados de las pruebas computacionales, se evidencia que el enfoque bidireccional de la heurística JOMI se destaca notablemente, ya que generó el mejor promedio de eficiencia de los algoritmos testeados con un 91,42%, además de obtener en el 69,52% de los casos (187 de las 269 instancias) la solución óptima para dicha instancia. También se pudo verificar que este procedimiento puede ser utilizado a cabalidad en casos reales. Gracias a los resultados observados, se puede decir que se logró el objetivo principal de la investigación, el cual era desarrollar un método alternativo para solucionar de manera eficiente los problemas de equilibrio de líneas de ensamble.

El algoritmo propuesto puede ser de gran ayuda en el diseño y desarrollo de nuevos algoritmos para la solución de los SALBP-1; además, puede ser utilizado en casos reales y en problemas generales como es el caso de problemas de líneas de ensamble con incompatibilidad de tareas, balanceo de líneas de ensamble en forma de U, problemas en paralelo, entre otros. Por tanto, se recomienda a estudiosos del tema que se fundamenten en este algoritmo para el desarrollo de nuevas investigaciones.

REFERENCIAS

- Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G.C. & Stewart, W. R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1, 9-32. doi: 10.1007/BF02430363
- Bautista, J., Fernández, E., Gonzalez, J.L. & Laguna, M.. (Septiembre, 2005). Hiperheurística para un problema de equilibrado de líneas de montaje usando Scatter Search. En F. Herrera & J.J. Merelo (presidencia), *Congreso español sobre metaheurísticas, algoritmos evolutivos y bioinspirados*. Simposio llevado a cabo en el IV Congreso español sobre metaheurísticas, algoritmos evolutivos y bioinspirados, Universidad de Granada, Granada, España.
- Bautista, J. & Pereira, J. (2007). Ant algorithms for a time and space constrained assembly line balancing problem. *European journal of operational research*, 177. 2016-2032. doi: 10.1016/j.ejor.2005.12.017
- Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management science*, 32. 909-932. doi: 10.1287/mnsc.32.8.909
- Becker, C. & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168. 694-715. doi: 10.1016/j.ejor.2004.07.023
- Becker, C. & Scholl, A. (2009). Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. *European Journal of Operational Research*, 199. 359-374. doi:10.1016/j.ejor.2008.11.051
- Boysen, N., Fliedner, M. & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research*, 183. 674-693. doi: 10.1016/j.ejor.2006.10.010
- Burgos, J., Burgos, J., Garzón, L. & López L. (2013). JOMI: Algoritmo heurístico tipo greedy para la solución de los problemas de líneas de ensamble. *DYNA Management*, 1. p. [no data] doi: 10.6036/MN5714
- Chica, M., Cordon, Ó., Damas, S. & Bautista, J. (2010). Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Information Sciences*, 180. 3465-3467. doi: 10.1016/j.ins.2010.05.033
- Feo, T. A. & Resenden, M. C.G. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6. 109-133.
- Hooker, J.N. (1995). Testing heuristics: we have it all wrong. *Journal of Heuristics*, 1. 33-42. doi: 10.1007/BF02430364
- Johnson, R.V. (1988). Optimally balancing large assembly line witch "FABLE". *Management science*, 34. 240-253. doi: 10.1287/mnsc.34.2.240
- Kilincici, O. (2011). Firing sequences backward algorithm for simple assembly line balancing problem of type 1. *Computers & Industrial Engineering*, 60. 830-839 doi: 10.1016/j.cie.2011.02.001

- Krzysztof, F. & Khalil, S. (2003). An enumerative heuristic and reduction methods for the assembly line balancing problem. *European journal of operational research*, 145. 606-620. doi: 10.1016/S0377-2217(02)00204-7
- Lapierre, S.D., Ruiz, A. & Soriano, P. (2006). Balancing assembly lines with tabu search. *European journal of operational research*, 168. 826- 837. doi: 10.1016/j.ejor.2004.07.031
- Pachghare, V. & Dalu R. (2012). Assembly Line Balancing Methods—A Case Study. *International Journal of Science and Research (IJSR)*, 3.1901-1905. ISSN: 2319-7064
- Rardin, R.L. & Uzsoy, R. (2001). Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics*, 7. 261-304 doi: 10.1023/A:1011319115230
- Rubinovitz, J. & Levitin, G. (1995). Genetic algorithm for assembly line balancing. *International Journal of Production Economics*, 41. 343-354. doi:10.1016/0925-5273(95)00059-3
- Sabuncuoglu, I., Erel, E. & Alp, A. (2009). Ant colony optimization for the single model U-type assembly line balancing problem. *Int. J. Production Economics*, 120. 287-300. doi: 10.1016/j.ijpe.2008.11.017
- Scholl A. (1999). *Balancing and sequencing of assembly lines*. Heidelberg, Alemania: Physica-Verlag.
- Scholl, A. & Becker, C. (2006). State-Of-The-Art Exact and Heuristic Solution Procedures For Assembly Line Balancing. *European Journal of Operational Research*, 168. 666-693. doi:10.1016/j.ejor.2004.07.022.
- Scholl, A. & Klein, R. (1997). SALOME: a bidirectional branch and bound procedure for assembly line balancing. *INFORMS journal on Computing*, 9. 319-334. doi: 10.1287/ijoc.9.4.319
- Scholl, A. & Voss, S. (1996). Simple assembly line balancing – Heuristic approaches. *Journal of Heuristics*, 2. 217-244. doi: 10.1007/BF00127358
- Silver, E.A. (2004). An overview of heuristic solution methods. *Journal of the Operational Research Society*, 55. 936-956 doi: 10.1057/palgrave.jors.2601758
- Talbot, F.B., Patterson, J.H. & Gehrlein, W.V. (1986). A comparative evaluation of heuristic line balancing Techniques. *Management science*, 32. 430-454. doi: 10.1287/mnsc.32.4.430
- Tuncel, G. & Topaloglu, S. (2013). Assembly line balancing with positional constraints, task assignment restrictions and station paralleling: A case in an electronics company. *Computers & Industrial Engineering*, 64. 602-609. doi: 10.1016/j.cie.2012.11.006
- Yeh, D-H. & Kao, H-H. (2009). A new bidirectional heuristic for the assembly line balancing problem. *Computers & Industrial Engineering*, 57. 1155-1160. doi:10.1016/j.cie.2009.05.004