

## PROBLEMAS DE BALANCEO DE LÍNEA SALBP-1 Y SALBP-2: UN CASO DE ESTUDIO

### Assembly balancing line problem SALBP-1 and SALBP-2: a case of study

#### RESUMEN

Este documento presenta el problema de balanceo de línea de ensamble SALBP-1 y SALBP-2. El problema se resuelve con el algoritmo COMSOAL (Computer Method for Sequencing Operations for Assembly Lines) implementado en Visual Basic para Excel.

**PALABRAS CLAVES:** Balanceo de líneas de ensamble, SALBP-1, SALBP-2

#### ABSTRACT

This paper shows assembly balancing line problem SALBP-1 and SALBP-2. The problems are solved with COMSOAL algorithm (Computer Method for Sequencing Operations for Assembly Lines). It is implemented in Visual Basic for Excel.

**KEYWORDS:** Assembly balancing line problem, SALBP-1, SALBP-2

#### JORGE HERNAN RESTREPO

Ingeniero Industrial, M. Sc.  
Profesor Asistente  
Universidad Tecnológica de Pereira  
jhrestrepoco@utp.edu.co

#### PEDRO DANIEL MEDINA V

Ingeniero Mecánico, M. Sc.  
Profesor Especial  
Universidad Tecnológica de Pereira  
[pemedin@utp.edu.co](mailto:pemedin@utp.edu.co)

#### EDUARDO ARTURO CRUZ T

Ingeniero Industrial, M. Sc.  
Profesor Asistente  
Universidad Tecnológica de Pereira  
[ecruz@utp.edu.co](mailto:ecruz@utp.edu.co)

## 1. INTRODUCCIÓN

En un mundo cada vez más competitivo, en donde el mercado actual es cada vez más exigente, las empresas modernas se ven obligadas a evolucionar en todos sus aspectos funcionales. En la actualidad, por efecto de la globalización, las industrias deben mejorar la competitividad, y esto conlleva el mejoramiento de los procesos de manufactura, es decir la optimización de las variables y recursos de la empresa, para reducir costos, mejorar calidad y eficiencia; por ello adquiere importancia el estudio del balanceo de línea de ensamble.

Podemos observar que la eficiencia productiva juega un papel muy importante en el crecimiento de las empresas, es por tal razón que se debe procurar por el buen uso de los recursos en el proceso de producción, de tal manera que las empresas deben preocuparse por conocer, estudiar y mejorar el comportamiento de las diferentes variables involucradas al interior de su sistema de manufactura, específicamente, la línea de ensamble, ya que es aquí en donde se presentan diferentes problemas: minimizar estaciones, minimizar tiempo ocioso, maximizar flujo de producción, minimizar tiempos de ciclo; lo cual genera en la empresa una disminución en los costos de operación y en consecuencia, aumento en la competitividad. El estudio de los sistemas de producción y operaciones, es muy importante, dentro de la ingeniería industrial, que requiere en la actualidad la solución de problemas que presentan las empresas en el área de producción, utilizando diversas herramientas y métodos para

solucionarlos. Esta investigación se centrará en el balanceo de una línea de ensamble simple, la cual es la línea más común en la industria moderna.

## 2. TEORÍA [1][2]

### EL PROBLEMA DEL EQUILIBRADO DE LÍNEAS DE MONTAJE

El problema del equilibrado de líneas de ensamble o de montaje (en inglés: assembly line balancing problem) es un problema actual en el mundo industrial. Se trata de un problema que ha sido tratado de forma constante a lo largo de la historia por diferentes autores; y que en la actualidad aún es objeto de estudio por un gran número de investigadores. A continuación se muestra cuál ha sido la evolución histórica del problema, así como los diferentes tipos de problemas de equilibrado de líneas que existen.

Una línea de ensamblaje es una secuencia de estaciones de trabajo, que están frecuentemente conectadas por un sistema de manejo de materiales, la cual existe con el propósito de ensamblar un producto final. El proceso consiste en una serie de operaciones, cada una compuesta por diversas tareas. Una tarea consiste en un conjunto de elementos básicos e indivisibles en forma lógica.

El problema de equilibrado de líneas de montaje puede ser de dos tipos: Simple (Simple Assembly Line

Balancing Problem, SALBP) o General (General Assembly Line Balancing Problem, GALBP).

### **SALBP: Simple Assembly Line Balancing Problem**

El SALBP presupone que todos los parámetros se conocen con certeza, así, una tarea no puede ser dividida entre dos o más estaciones de trabajo, existen secuencias tecnológicas que deben respetarse (que determinan relaciones de precedencia e incompatibilidad entre las tareas) y todas las tareas deben llevarse a cabo. También se presupone que todas las estaciones de trabajo están equipadas para poder realizar cualquier tarea y bajo el mismo costo. A la vez, la duración de realización de una tarea es independiente de la estación de trabajo a la que sea asignada y de las tareas que la hayan precedido en ella.

Teniendo en cuenta las anteriores características comunes, no obstante, se pueden distinguir cuatro casos distintos de SALBP:

a) SALBP-1: se caracteriza por disponer de un tiempo de ciclo asignado (o tasa de producción) y desea minimizar el número de estaciones que se requiere para llevar a cabo el proceso. Es muy común que se de cuando la demanda externa puede ser estimada y un nuevo sistema de montaje vaya a ser instalado.

b) SALBP-2: a diferencia del caso anterior, se parte de un número de estaciones fijado y se desea minimizar el tiempo de ciclo de la línea de montaje en cuestión. Se acostumbra a presentar dicho problema cuando la línea de montaje ya existe.

c) SALBP-E: se busca maximizar la eficiencia de la línea de montaje, o lo que es equivalente, se desea minimizar el producto del número de estaciones y el tiempo de ciclo.

d) SALBP-F: en este caso no se busca maximizar ni minimizar ningún valor sino que se limita a encontrar, si existe, una solución factible para una combinación cualquiera de tiempo de ciclo y número de estaciones.

### **GALBP: General Assembly Line Balancing Problem**

Esta tipología de problemas, los GALBP, engloban a todos aquellos problemas de equilibrado de líneas de montaje que no son SALBP. Los más comunes son los que tienen, por ejemplo: modelos mixtos, estaciones en paralelo, procesamientos alternativos, tiempos de proceso variables; se caracterizan por dar cabida a problemas más reales y cotidianos que los SALBP. Se destacan cuatro tipos diferentes de problemas GALBP:

a) UALBP (U-line Assembly Line Balancing Problem): se caracterizan por, en vez de tratar con una línea serial, trabajar con una de tipo U. En una línea de tipo U las

estaciones pueden ser colocadas de tal manera que se pueden manejar a la vez dos piezas en diferentes posiciones de la línea. Como se puede consultar en diferentes ejemplos, en esta tipología de línea hay un mayor número de posibilidades de asignar las tareas a las estaciones, con lo que el problema se puede resolver de manera más eficiente que cuando se tiene una línea serial. También se pueden distinguir: UALBP-1 (minimizar el número de estaciones), UALBP-2 (minimizar el tiempo de ciclo) y UALBP-E (maximizar la eficiencia de la línea).

b) MALBP (Mixed-Model Assembly Line Balancing Problem): equilibrado de líneas con modelos mixtos; se caracterizan por tener varios modelos de un mismo producto, teniendo un conjunto de tareas común a realizar en la totalidad de los modelos y sin tener en cuenta tiempos de preparación de la línea. Al igual que en las tipologías anteriores, también se presentan problemas MALBP-1, MALBP-2 y MALBP-E.

c) RALBP (Robotic Assembly Line Balancing Problem): equilibrado de líneas robotizadas; se desea optimizar la ejecución de las tareas en la línea, considerando tanto la asignación de las tareas a cada una de las estaciones como la destinación de cada uno de los robots a las diferentes estaciones.

d) MOALBP: Multi-Objective Assembly Line Balancing Problem: equilibrado de líneas con objetivos múltiples; se caracterizan por buscar varios objetivos a la vez combinando, por ejemplo, minimizar el número de estaciones, el costo o el número de pulmones, y maximizar la eficiencia. La mayoría de los problemas de equilibrado de líneas buscan cumplir objetivos múltiples. Una vez realizada esta primera clasificación, se tiene que tener en cuenta que, a su vez, los dos tipos de problemas tratados (SALBP y GALBP), se pueden subdividir teniendo en cuenta dos conceptos más:

- El tipo de producto que se procesa en la línea: modelo simple y modelo mixto (o múltiple)
- La variabilidad de las duraciones de las tareas: determinista o estocástico.

Para resolver el problema SALBP hay métodos exactos y heurísticos, siendo uno de ellos el COMSOAL (Computer Method for Sequencing Operations for Assembly Lines)[3]. El método se expone a continuación para dar solución al problema planteado en este documento.

### **3. PLANTEAMIENTO DEL PROBLEMA**

En este trabajo se resuelve un problema de 57 tareas con el algoritmo COMSOAL[4]. Se considerará el problema como SALBP-1 y SALBP-2. Se asume un tiempo de ciclo de 100 unidades de tiempo y 6 estaciones de trabajo

para cada uno de los casos. A continuación se presentan los modelos matemáticos para cada uno de los casos.

### 3.1 MODELO MATEMÁTICO[5] PARA EL SALB-1

$$MinZ = \sum_{j=1}^{M_{max}} y_j$$

s.a

$$\sum_{i=1}^N t_i \cdot x_{ij} \leq C \cdot y_j \quad j=1, \dots, M_{max} \quad (1)$$

$$\sum_{j=1}^{M_{max}} x_{ij} = 1 \quad i=1, \dots, N \quad (2)$$

$$\sum_{j=1}^{M_{max}} j \cdot x_{kj} \leq \sum_{j=1}^{M_{max}} j \cdot x_{ij} \quad \forall k < i \quad (3)$$

$$y_{j+1} \leq y_j \quad j=1, \dots, M_{max}-1 \quad (4)$$

$$x_{ij} = \{0,1\} \quad \forall(i,j); \quad y_j = \{0,1\} \quad \forall(j)$$

$x_{i,j} = 1$ , Si la tarea i se hace en la estación j

$y_j = 1$ , Si existe la estación j

$t_i$  = tiempo de proceso de la tarea i

C = Tiempo de ciclo

- (1) Duración en cada estación, menor que C
- (2) Cada operación, a una estación
- (3) Cumpliendo relaciones de precedencia
- (4) Si una estación no existe, tampoco las siguientes

### 3.2 MODELO MATEMÁTICO[5] PARA EL SALB-2

Min C

s.a

$$\sum_{i=1}^N t_i \cdot x_{ij} \leq C \quad j = 1, \dots, M \quad (1)$$

$$\sum_{j=1}^M x_{ij} = 1 \quad i = 1, \dots, N \quad (2)$$

$$\sum_{j=1}^M j \cdot x_{kj} \leq \sum_{j=1}^M j \cdot x_{ij} \quad \forall k < i \quad (3)$$

$$x_{ij} = \{0,1\} \quad \forall(i,j); \quad C \in \mathbb{R}^+$$

- (1) Duración en cada estación, menor que C
- (2) Cada operación, a una estación
- (3) Cumpliendo las relaciones de precedencia

### 3.3 ALGORITMO COMSOAL(Computer Method for Sequencing Operations for Assembly Lines)

El algoritmo tiene 6 pasos como siguen:

1. Para cada tarea, identifique las tareas que le siguen inmediatamente o preceden inmediatamente.
2. Crear una lista A con las tareas no asignadas, sus predecesoras inmediatas (no asignadas) y el número total de tareas que la preceden (no asignadas).
3. De la lista A cree una lista B conformada por las tareas que tienen cero (0) predecesores. Si no hay tarea sin ser asignadas a estaciones, entonces parar.
4. De la lista B, crear una lista C compuesta de las tareas que tienen tiempo de proceso que no supera el tiempo disponible de la estación. Si la lista C esta vacía, abra una nueva estación y repita el paso 2.
5. Aleatoriamente seleccione de la lista C una tarea para asignarla a la estación.
6. Actualice el tiempo disponible en la estación y la lista A. retorne al paso 3

### 4. EJEMPLO

Para realizar un producto se hacen 5 tareas. Ellas se presentan en el cuadro que sigue con sus tiempos y precedencias. Para C=8 determine el mejor balanceo. La tabla 1 presenta las tareas, tiempos de proceso y sus predecesores inmediatos.

Tarea	Tiempo	Predecesores
a	3	-
b	5	a
c	2	-
d	4	b,c
e	2	d

Tabla 1. Tareas, tiempos de proceso y sus predecesores inmediatos.

#### 4.1 APLICACIÓN DEL ALGORITMO COMSOAL

El algoritmo: Primera iteración:

1. Para cada tarea, identifique las tareas que le siguen inmediatamente o preceden inmediatamente.

Tarea	Tiempo	Predecesores
a	3	-
b	5	a
c	2	-
d	4	b,c
e	2	d

2. Crear una lista A con las tareas no asignadas, sus predecesoras inmediatas (no asignadas) y el número total de tareas que la preceden (no asignadas)..

Lista A:

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0
b	5	a	1
c	2	-	0
d	4	b,c	2
e	2	d	1

3. De la lista A cree una lista B conformada por las tareas que tienen cero (0) predecesores. Sí no hay tarea sin ser asignadas a estaciones, entonces parar.

Lista B

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0
c	2	-	0

4. De la lista B, crear una lista C compuesta de las tareas que tienen tiempo de proceso que no supera el tiempo disponible de la estación. Sí la lista C esta vacía, abra una nueva estación y repita el paso 4.

Lista C

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0
c	2	-	0

5. Aleatoriamente seleccione de la lista C una tarea para asignarla a la estación.

Se selecciona la tarea C

6. Actualice el tiempo disponible en la estación y la lista A. retorne al paso 3

A la estación uno se asigna la tarea C con un tiempo de 2. El tiempo disponible es= Capacidad de la estación menos tiempo total asignado= 8-2= 6

El algoritmo: Segunda iteración:

1. Para cada tarea, identifique las tareas que le siguen inmediatamente o preceden inmediatamente.

Tarea	Tiempo	Predecesores
a	3	-
b	5	a
c	2	-
d	4	b,c
e	2	d

2. Crear una lista A con las tareas no asignadas, sus predecesoras inmediatas (no asignadas) y el número total de tareas que la preceden (no asignadas).

Lista A:

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0
b	5	a	1
d	4	b	1
e	2	d	1

3. De la lista A cree una lista B conformada por las tareas que tienen cero (0) predecesores. Sí no hay tarea sin ser asignadas a estaciones, entonces parar.

Lista B

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0

4. De la lista B, crear una lista C compuesta de las tareas que tienen tiempo de proceso que no supera el tiempo disponible de la estación. Sí la lista C esta vacía, abra una nueva estación y repita el paso 4.

Lista C

Tarea	Tiempo	Predecesores	Número Predecesores
a	3	-	0

5. Aleatoriamente seleccione de la lista C una tarea para asignarla a la estación.

Se selecciona la tarea A

6. Actualice el tiempo disponible en la estación y la lista A. retorne al paso 3

A la estación uno se asigna la tarea A con un tiempo de 3. El tiempo disponible es= Capacidad de la estación menos tiempo total asignado= 8-5= 3

El algoritmo: tercera iteración:

1. Para cada tarea, identifique las tareas que le siguen inmediatamente o preceden inmediatamente.

Tarea	Tiempo	Predecesores
a	3	-
b	5	a
c	2	-
d	4	b,c
e	2	d

2. Crear una lista A con las tareas no asignadas, sus predecesoras inmediatas (no asignadas) y el número total de tareas que la preceden (no asignadas).

Lista A:

Tarea	Tiempo	Predecesores	Número Predecesores
b	5	-	0
d	4	b	1
e	2	d	1

3. De la lista A cree una lista B conformada por las tareas que tienen cero (0) predecesores. Si no hay tarea sin ser asignadas a estaciones, entonces parar.

Lista B

Tarea	Tiempo	Predecesores	Número Predecesores
b	5	-	0

4. De la lista B, crear una lista C compuesta de las tareas que tienen tiempo de proceso que no supera el tiempo disponible de la estación. Si la lista C esta vacía, abra una nueva estación y repita el paso 2.

Lista C

Tarea	Tiempo	Predecesores	Número Predecesores
-	-	-	-

La única tarea que podría pasar a la lista C es la tarea B que tiene un tiempo de 5, pero supera el tiempo disponible de la estación que es 3, por tanto se debe abrir otra estación y retornar al paso 2.

Hasta el momento esta secuencia de asignación se ha generado 2 estaciones. Repitiendo el algoritmo hasta ser asignadas todas las tareas a una estación. Al final se tiene el siguiente resultado:

Estación 1: c, a. tiempo asignado=5

Estación 2: b, tiempo asignado=5

Estación 3: d, e. tiempo asignado=6

Con el COMSOAL se pueden tener diferentes secuencias cada vez que el se ejecuta. La anterior secuencia es el resultado de la selección aleatoria de la tarea C, como primera tarea a ser asignada.

### 5. DATOS DEL PROBLEMA

El problema se resuelve para el SALB-1 con un C=100 y para el problema SALBP-2 con 6 estaciones.

La tabla 2 presenta las tareas, predecesores y tiempos de proceso.

Tarea	Predecesores	Tiempo Proceso
1		7
2	1	10
3		4
4	3	6
5	4	11
6	2,5	6
7	6	7
8	7	10
9	7	8
10	7	3
11	2,5	15
12	8,9,10,11	9
13	2,5	8
14	12	12
15	13,14	5
16	15	14
17	2,5	6
18	2,5	7
19	2,5	10
20	2,5	8
21	2,5	4
22	16	10
23	16	7
24	23	22
25	22	11
26	2,5	16
27	24,25	6
28	27	12
29	2,5	15
30	17,18,19,20	9
31	30	8
32	31	8

33	32	9
34	32	5
35	33,34	18
36	29,35	7
37	36	12
38	29,35	14
39	29,35	10
40	29,35	5
41	37,38,39,41	11
42	37,38,39,40	10
43	41,42	10
44	43	8
45	43	8
46	29,35	6
47	29,35	15
48	44,45,46	15
49	29,35	7
50	47,48,49	7
51		8
52	51	8
53	52	11
54		6
55	54	8
56	55	8
57	53	2

Tabla 2. Tarea, predecesores y tiempos de proceso.

**5. RESULTADOS**

Para el balanceo de la línea con C=100 se tiene la asignación que se relaciona en la tabla 3. Esta asignación genera una eficiencia de línea del 90.6%

1	2	3	4	5	6
Asignación Estaciones					
3	17	54	23	32	47
4	18	51	22	33	39
1	9	52	25	34	42
5	26	53	19	35	41
2	8	56	24	36	43
11	12	57	27	40	44
6	20		28	37	45
13	14		30	38	48
7	15		31	46	50
10	16			49	
29					
21					

Tabla.3 Asignación de tareas con C=100

Para el balanceo de la línea con M=6 se tiene la asignación que se relaciona en la tabla 4. Esta asignación genera una eficiencia de línea del 90.6% y un tiempo de ciclo C=99 en la segunda estación.

1	2	3	4	5	6
Asignación Estaciones					
51	3	9	26	32	38
52	1	8	22	33	39
53	4	12	25	34	41
54	5	14	24	35	42
57	2	15	27	40	43
55	20	17	28	46	45
56	13	19	30	49	44
	11	16	31	36	48
	21	29		37	50
	18	23		47	
	6				
	7				
	10				

Tabla 4. Asignación de tareas con M=6

**6. CONCLUSIÓN**

El algoritmo es fácil de implementar y permite encontrar varias soluciones a diferencia de otros procedimientos como la heurística de pesos posicionales. Se recomienda comparar este algoritmo con otros procedimientos como las metaheurísticas para observar sus debilidades o fortalezas.

**7. BIBLIOGRAFÍA**

[1] Nils Boysen, Malte Fliedner, Armin Scholl. A classification of assembly line balancing problems, European Journal of Operational Research, Volume 183, Issue 2, 1 December 2007, Pages 674-693.

[2] Capacho B Liliana, Moreno Rafael P. Generación de secuencias de montaje y equilibrado de líneas. Doctorado automatización avanzada y robótica, Universidad politécnica de Cataluña, Barcelona España 2004.

[3] Armin Scholl, Christian Becker, State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, European Journal of Operational Research, Volume 168, Issue 3, 1 February 2006, Pages 666-693.

[4] Askin Ronald, Standridge Charles. Modeling and analysis of manufacturing systems. Assembly lines: Reliable serial systems. Page 31. edit wiley 1993.

[5] Eguía Ignacio. Equilibrado de líneas, Escuela Superior de Ingenieros, Universidad de Sevilla, Dpto. Organización Industrial y Gestión de Empresas 2008.