**SOLUCIONES DEL PROBLEMA DE BALANCEO DE LÍNEA SALBP-2. UN ENFOQUE BOOLEANO CON PROGRAMACIÓN EN PARALELO.**

**RESUMEN**

Se presenta un algoritmo para encontrar todas las soluciones del problema de Balanceo de Línea SALBP-2. El algoritmo parte de un ciclo ideal y de la generación de restricciones booleanas para este ciclo. Las restricciones son manipuladas lógicamente, hasta que se encuentran el espacio de soluciones. Si el espacio booleano generado es vacío, se incrementa el ciclo ideal con el máximo común divisor y se repite el ciclo, hasta encontrar todas las soluciones con el ciclo óptimo. Además de encontrar todas las soluciones, el algoritmo puede paralelizarse con relativa facilidad, para ejecutarse en ambientes CUDA C. Los tiempos de ejecución con problemas benchmark son relativamente buenos.

<https://docs.google.com/presentation/d/1QFajvdVtlx-0lgN-7gol41Sk_bMVp9cwIbEXb-fNUUM/edit?ts=5aa965fd#slide=id.p3>

**1 DEFINICIÓN DEL PROBLEMA**

**1.1 Línea de Ensamble**

Un sistema de ensamble tiene como objetivo el ensamble de algún producto o productos mediante la realización de unidades indivisibles de trabajo, llamadas *actividades*.

Las actividades se realizan en *estaciones de trabajo*. El sistema está formado por un mecanismo de transporte que une a las estaciones de trabajo; además, se especifica el orden de las estaciones y de las actividades que debe seguir el ensamble del producto. Se conoce como *línea de ensamble* al conjunto de estaciones y al mecanismo que transporta el ensamble.

Se llama *tiempo de ciclo c* al intervalo de tiempo que transcurre antes de que el mecanismo de transporte se mueva. Se denomina como *tiempo de procesamiento* al tiempo necesario para realizar una actividad. El *contenido de trabajo o duración* de una estación se obtiene sumando todos los tiempos de procesamiento de las actividades que se realizan en esa estación. Para evitar retrasos en la línea, el contenido de trabajo de las estaciones debe ser menor que el intervalo de tiempo en el cual se mueve el mecanismo de transporte.

Se le llama *holgura* al tiempo de ocio con el que cuenta cada estación, i.e., el tiempo de ciclo menos su duración. Si la holgura de cada una de las estaciones es mayor que cero entonces una unidad se termina de ensamblar cada tiempo *c*, es decir, la tasa de producción de la línea es igual a 1/*c*. Como ya se estableció, el orden en el cual se realizan las actividades está especificado y se llaman *relaciones de precedencia* a la especificación de este orden. Debido a estas relaciones de precedencia, no se puede comenzar una actividad hasta que todos sus predecesores inmediatos hayan sido terminados. Se pueden representar gráficamente estas relaciones a través de un *grafo de precedencias*. En este trabajo, se supone que la línea de ensamble no tiene reprocesos o retroalimentaciones, por lo tanto el grafo que se obtiene no es cíclico.

Si cada nodo del grafo simboliza a una actividad entonces, para representar una relación de precedencia inmediata de una actividad *i* a una actividad *j*, se utiliza un arco dirigido desde el nodo *i* hacia el nodo *j*.

Si la suma de las holguras de todas las estaciones es la mínima posible, entonces la línea de ensamble se encuentra *balanceada* [Baybars, 1986].

**1.2 Una Definición del Problema Balanceo de Línea de Ensamble Simple**

En términos generales el problema de balanceo de línea (ALBP por sus siglas en inglés) consiste en asignar las actividades en la línea de ensamble a un conjunto de estaciones optimizando alguna variable objetivo sin violar ciertas restricciones. Dependiendo del objetivo de optimización y de las restricciones, se derivan varios modelos de ALBP [Baybars1986].

A continuación se enumeran los supuestos que [Baybars, 1986] establece, sobre los cuales se basa la versión simple de ALBP (SALBP por sus siglas en inglés):

S-1 Todos los parámetros de entrada se conocen con certeza.

S-2 Una actividad no se puede dividir entre dos o más estaciones.

S-3 Las actividades no pueden ser procesadas en secuencias arbitrarias debido a ciertos requerimientos tecnológicos de precedencia.

S-4 Todas las actividades deben ser procesadas.

S-5 Todas las estaciones en consideración están equipadas, en cuanto a máquinas y trabajadores, para procesar cualquiera de las actividades.

S-6 Los tiempos de procesamiento de las actividades son independientes de la estación en la que se llevan a cabo o de sus predecesoras o sucesoras.

S-7 Cualquier actividad puede ser procesada en cualquier estación.

S-8 La línea es considerada en su totalidad como serial, es decir, no existen líneas de ensamble secundarias o líneas paralelas de sub-ensamble.

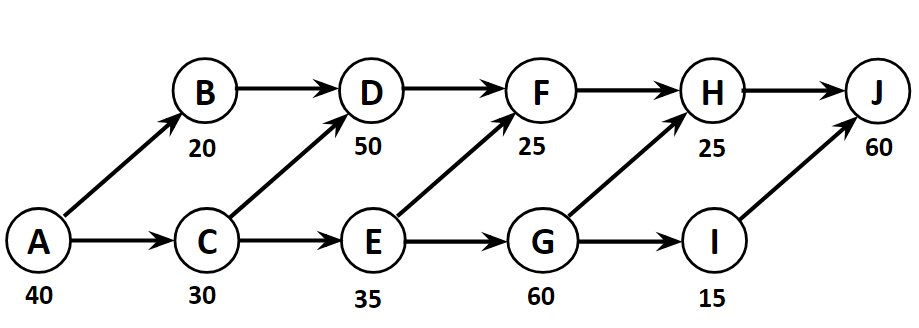
S-9 Se asume que el sistema de ensamble está designado para un único modelo de un único producto.

El objetivo del tipo general de SALBP, SALBP-G, es buscar el número de estaciones *m* y el tiempo de ciclo *c* que minimizan la holgura total del sistema. El problema SALBP-G es de tipo no lineal, por lo que surgen dos versiones lineales simplificadas de ese problema. La versión SALBP-1 consiste en minimizar el número de estaciones *m* dado un tiempo de ciclo *c*, en tanto que la versión SALBP-2 consiste en minimizar el tiempo de ciclo *c* dado un número de estaciones *m*.

El procedimiento aquí propuesto se centra en solucionar la versión SALBP-2. En específico, se busca una asignación de las actividades de las actividades a las estaciones de trabajo tal que se minimice el tiempo de ciclo *c*.

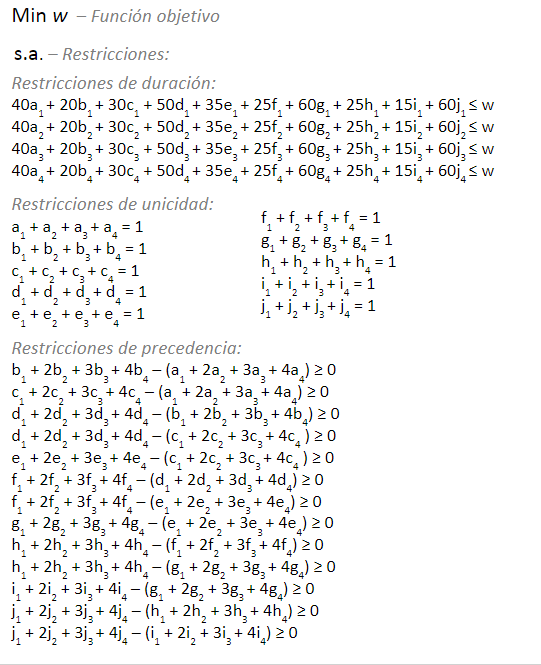
**1.2 Ejemplo para Ilustrar el Método Propuesto**

Para ilustrar el método aquí propuesto, se usará la siguiente línea de producción:



Esta línea tiene 10 actividades, un tiempo total de 360 minutos y se quiere encontrar el ciclo óptimo con cuatro estaciones de trabajo.

Usando Programación Lineal Mixta, este problema podría ser resuelto mediante la siguiente formulación:



Todas las variables del modelo, con excepción de w, son binarias. Por ejemplo, dentro de las restricciones de unicidad En palabras, el problema consiste en minimizar la máxima w, que en este caso representa el ciclo. (ciclo)

Baybars, İlker. «A Survey of Exact Algorithms for the Simple Assembly Line BalancingProblem.» Management Science 32, nº 8 [August 1986]: 909-932.

Boysen, Nils, Malte Fliedner, y Armin Scholl. «A classification of assembly line balancing

problems.» *European Journal of Operational Research* 183 [2007]: 674-693.

Scholl, Armin. «Data of Assembly Line Balancing Problems.» *Schriften zur Quantitativen*

*Betriebswirtschaftslehre* [Technische Universität Darmstadt] 16/93 [November

1993]: 1-30.

Scholl, Armin, Malte Fliedner, y Nils Boysen. «ABSALOM: Balancing assembly lines with

assignment restrictions.» *European Journal of Operational Research* 200 [2009]:

688-701.