

## Enfoque híbrido metaheurístico AG-RS para el problema de asignación del buffer que minimiza el inventario en proceso en Líneas de producción abiertas en serie

José Omar Hernández-Vázquez<sup>a,\*</sup>, Salvador Hernández-González<sup>b</sup>, José Alfredo Jiménez-García<sup>b</sup>  
Manuel Darío Hernández-Ripalda<sup>b</sup>, José Israel Hernández-Vázquez<sup>a</sup>

<sup>a</sup> Departamento de Doctorado en Ciencias de la Ingeniería, Tecnológico Nacional de México en Celaya

<sup>b</sup> Departamento de Ingeniería Industrial, Tecnológico Nacional de México en Celaya

### Resumen

El problema de asignación del buffer (BAP, por sus siglas en inglés) es clasificado como un problema de optimización combinatorio NP-Duro en el diseño de las líneas de producción. Éste consiste en definir la asignación de lugares de almacenamiento (buffers) dentro de una línea de producción, con el fin de aumentar al máximo la eficiencia del proceso. Los métodos de optimización que han sido reportados con mayor éxito en los últimos años son las técnicas metaheurísticas. En este trabajo, se propone un enfoque híbrido que utiliza las técnicas metaheurísticas de: Algoritmos Genéticos (AG) y Recocido Simulado (RS), con el objetivo de determinar los buffers requeridos que minimicen el promedio de inventario en proceso (WIP, por sus siglas en inglés) en líneas de producción abiertas en serie  $M/M/1/K$ . La evaluación se realiza con un método analítico de descomposición. Los resultados obtenidos demuestran la eficiencia computacional del algoritmo híbrido propuesto con respecto a un RS o AG estándar.

### Palabras Clave:

Control de inventario, Optimización y métodos computacionales, BAP, Metaheurísticas híbridas, Líneas de producción.

### Hybrid metaheuristic approach GA-SA for the buffer allocation problem that minimizes the work in process in open serial production lines

### Abstract

The Buffer Allocation Problem (BAP) is a problem of combinatorial NP-Hard optimization in the design of production lines. This consists of defining the allocation of storage places (buffers) within a production line, in order to maximize the efficiency of the process. The methods of optimization have been reported with greater success in recent years are metaheuristic techniques. In this work, a hybrid approach is proposed that uses the metaheuristic techniques of Genetic Algorithms (GA) and Simulated Annealing (SA), with the objective of determining the required buffers that minimize the average work in process (WIP) in open serial production lines  $M/M/1/K$ . The evaluation is carried out with an analytical method of decomposition. The results obtained demonstrate the computational efficiency of the proposed hybrid algorithm with respect to a simple SA or GA.

### Keywords:

Inventory control, Optimization and computational methods, BAP, Hybrid metaheuristics, Production lines.

## 1. Introducción

La principal razón para mantener lugares de almacenamiento (buffers) en cualquier proceso de producción, es permitir que las máquinas secuenciales operen de forma independiente entre unas y otras. Los buffers reducen el tiempo de inactividad debido a la falta de abastecimiento y el bloqueo generado entre ellas. Menos tiempo de inactividad aumenta la producción promedio de la línea de fabricación. Por otro lado, la inclusión de buffers genera costos y requiere de un espacio

adicional de trabajo, por lo que puede ser inadecuada en algunos casos. Si el número de buffers es muy grande, los costos directos por capital, costos indirectos por incremento en el inventario en proceso (WIP) y los tiempos de ciclo generados serán mayores a los beneficios obtenidos por la productividad. Si los buffers colocados son pocos, será complicado aprovechar al máximo la capacidad de las máquinas, o bien, se corre el riesgo de no cumplir con la demanda (Kose and Kilincci, 2015). Debido a la importancia de encontrar las mejores configuraciones de buffers (u

\*Autor para la correspondencia: d1703005@itcelaya.edu.mx

óptimas), el problema de asignación del buffer (BAP, por sus siglas en inglés) es un factor muy importante a considerar por cualquier persona encargada de diseñar líneas de producción. El BAP ha sido estudiado por más de 50 años y numerosas publicaciones sobre este tema son posibles encontrar en la literatura (Demir *et al.*, 2014), (Weiss *et al.* 2018).

En la actualidad, la implementación de diversas estrategias de optimización ha generado resultados de un impacto positivo en diferentes campos de la ingeniería, basta con mencionar algunos ejemplos como son: técnicas de optimización multiobjetivo tanto para la identificación de parámetros de un modelo no lineal como para el ajuste de controladores (Huilocapi *et al.*, 2018), optimización en tiempo real a través de la metodología de adaptación de modificadores (Rodríguez-Blanco *et al.*, 2018) y el empleo de Algoritmos Genéticos para optimizar el control adaptativo fraccionario en reguladores automáticos de voltaje (Ortiz-Quisbert *et al.*, 2016). Esta última estrategia de optimización pertenece al grupo de las llamadas *técnicas metaheurísticas*.

Recientemente en el estudio del BAP, las técnicas metaheurísticas han reportado con mucho éxito soluciones de mayor calidad en tiempos relativamente cortos. Algunas de las técnicas utilizadas son: Búsqueda Tabú (Costa *et al.*, 2015), Recocido Simulado (Spinellis and Papadopoulos, 2000a, b), (Spinellis *et al.*, 2000), Algoritmos Genéticos (Spinellis and Papadopoulos, 2000b), (Cruz *et al.*, 2012), (Nahas *et al.*, 2014), (Nahas and Nourelfath, 2018) Colonia de Hormigas (Nahas *et al.*, 2009), (Nahas and Nourelfath, 2018) y Optimización por Enjambre de Partículas (Narasimhamu *et al.*, 2014, 2015).

Para la mejor exploración del espacio de soluciones del BAP, una propuesta reciente es la hibridación de técnicas metaheurísticas con otros métodos tales como: Particiones Anidadas (Shi and Men, 2003), métodos de Ramificación y Acotamiento (Dolgui *et al.*, 2007), Método del Subgradiente (Demir and Tunali, 2008), y Búsqueda Local (Amiri and Mohtashami, 2012), (Mohtashami, 2014). Existen también estudios que reportan la hibridación entre técnicas metaheurísticas, (Kose and Kilincci, 2015) hibridan Algoritmos Genéticos con Recocido Simulado con el fin de maximizar la tasa promedio de producción; por su parte (Su *et al.*, 2017) resuelven el BAP bajo un criterio multiobjetivo en donde se busca maximizar la tasa promedio de producción a la vez que se minimiza el total de buffers asignados, se aplica la hibridación de Búsqueda Tabú y NSGA-II; finalmente (Kose and Kilincci, 2018) resuelven el problema bajo un criterio multiobjetivo similar al empleado en el estudio anterior, a través de hibridar NSGA-II y Recocido Simulado. Los trabajos mencionados reportan resultados en muchas ocasiones más eficientes, por lo que el enfoque de hibridación parece ser un área de investigación muy prometedora.

En este estudio, se presenta una nueva propuesta de solución del BAP basada en un algoritmo híbrido metaheurístico que emplea Algoritmos Genéticos y Recocido Simulado. Últimamente, esta forma de hibridación ha permitido alcanzar resultados satisfactorios en el estudio de otros problemas clasificados como NP-Duro, por ejemplo: el problema de enrutamiento de vehículos con múltiples viajes (Ariyani *et al.*, 2018) y el problema de programación de órdenes de fabricación (Wei *et al.*, 2018). A diferencia de la mayoría de los trabajos reportados en la literatura sobre el BAP, los cuales buscan *maximizar la tasa promedio de*

*producción o throughput*, el estudio tiene como objetivo *minimizar el promedio de inventario en proceso (WIP Promedio)*. Menciona (Demir *et al.*, 2014) que el BAP bajo este criterio de optimización es el menos investigado, ya que implica analizar restricciones más desafiantes que otros modelos matemáticos. El empleo de un algoritmo híbrido con este criterio de optimización, resulta ser entonces una contribución diferente en el campo del BAP y la más importante del presente documento.

Una aportación relevante de este estudio, es la demostración de la potencia computacional del algoritmo híbrido AG-RS propuesto, con respecto a otros procedimientos metaheurísticos RS y AG estándar. Además, se hace una comparativa entre los algoritmos híbridos AG-RS y GAA, este último desarrollado por (Kose and Kilincci, 2015), los cuales revelan diferencias en sus criterios de optimización, parámetros y eficiencia computacional.

El resto de este documento está organizado de la siguiente manera. En la sección posterior se explica el BAP en líneas de producción en serie y lo que conlleva resolverlo. En la sección 3 se mencionan los sistemas markovianos. La descripción del algoritmo híbrido propuesto se detalla en la sección 4, mientras que la sección 5 señala el proceso de calibración de éste. En las secciones 6, 7 y 8 se muestran los resultados numéricos y estudios comparativos. Finalmente una sección de conclusiones es presentada, en ésta se abordan los alcances de los resultados generados y se establecen futuras direcciones de investigación.

## 2. El problema de asignación del buffer en líneas de producción en serie

### 2.1. Definición

El problema de asignación del buffer es clasificado como un problema de optimización combinatorio NP-Duro en el diseño de las líneas de producción (Demir *et al.*, 2014), (Weiss *et al.* 2018). Éste consiste en definir la asignación de lugares de almacenamiento (buffers) dentro de una línea de producción, con el fin de aumentar al máximo la eficiencia del proceso. En líneas de producción en serie, existen  $W$  máquinas con  $W-1$  áreas de almacén (denotadas por  $B_2, B_3, \dots, B_W$ ) entre dichas máquinas conectadas en serie, como se muestra en la Figura 1. El material fluye desde el exterior del sistema a  $W_1$ , luego a  $B_2$ , luego a  $W_2$ , y así sucesivamente hasta llegar a  $W_W$ , para después salir del proceso. En las líneas de producción abiertas, se considera que la primera máquina nunca sufrirá de escasez de material, y que la última máquina nunca estará bloqueada.

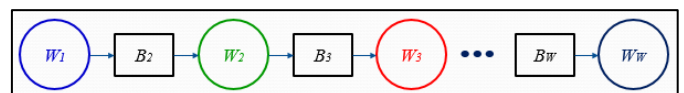


Figura 1: Línea de producción en serie con buffers.

La formulación del BAP puede ser expresada principalmente de tres formas distintas dependiendo de su función objetivo como lo muestran (Papadopoulos *et al.*, 2009) y (Demir *et al.*, 2014). La maximización de la tasa promedio de producción o throughput (BAP1), la minimización del tamaño total del buffer (BAP2) y la minimización del

promedio de inventario en proceso (BAP3). En este estudio se resuelve el modelo matemático BAP3 presentado en (Papadopoulos and Vidalis, 2001), el cual se muestra a continuación.

**Problema 3 (BAP3):** Esta formulación expresa la minimización del promedio de inventario en proceso (*WIP Promedio*), sujeto a un total de buffers disponibles y un nivel del throughput deseado.

$$\text{Min } Q(B) \quad (1)$$

Sujeto a:

$$\sum_{i=2}^W B_i = N \quad (2)$$

$$f(B) \geq f^* \quad (3)$$

$$B_i \geq 0 \quad (4)$$

Donde:

- $B = (B_2, B_3, \dots, B_W) =$  Vector del tamaño del buffer
- $Q(B) =$  WIP Promedio de la línea de producción en función del vector del tamaño del buffer
- $f(B) =$  Throughput de la línea de producción en función del vector del tamaño del buffer
- $f^* =$  Nivel del throughput deseado
- $N =$  Total de buffers disponibles
- $W =$  Máquinas en la línea de producción
- $B_i =$  Capacidad del buffer de la máquina  $i$  (variable entera no negativa)

## 2.2. Clasificación

Una clasificación del BAP es basada en conceptos de líneas cortas y largas. En (Papadopoulos et al., 2009) se establece que las *líneas cortas* pueden formarse con un máximo de seis estaciones y veinte buffers. Líneas con especificaciones fuera de estos rangos podrían ser consideradas como *líneas largas*. Cualquiera de estas líneas consiste en estaciones de una sola máquina que pueden estar *balanceadas* o *no balanceadas*. Por balanceada se entiende a una línea con iguales *tasas de servicios* ( $\mu$ ) o tiempos de procesamiento, en cada una de las  $W$  estaciones.

Otra característica de las líneas de producción es *confiables* y *poco confiables*. Una línea confiable es aquella en donde ninguna estación del proceso manifiesta fallas (o paros), es decir, el servidor único de cada estación (o máquina) siempre estará disponible para servir o producir inmediatamente a una parte que espera a ser procesada.

## 2.3. Procedimiento general para resolver el BAP

El procedimiento de solución del BAP consiste en un proceso de retroalimentación, entre un método de evaluación y un método generativo (o de optimización), que conduce a una solución óptima después de un número finito de iteraciones (Papadopoulos et al., 2009).

Los métodos de evaluación, que predicen las medidas de rendimiento del sistema, se basan en enfoques de agregación, métodos de descomposición, métodos de expansión o cualquier método aproximado, modelos Markovianos exactos y simulación. Por otra parte, los métodos generativos, que orientan la búsqueda de soluciones hacia valores óptimos, resultan ser variados. Sin embargo, en años recientes distintas técnicas metaheurísticas han sido adoptadas por investigadores para dar soluciones efectivas al BAP que posee en su naturaleza la esencia de los problemas combinatorios.

## 3. Sistemas markovianos con capacidad en la línea (buffer) M/M/1/K

El supuesto de capacidad infinita en una línea de espera difícilmente se cumple. Los sistemas markovianos designados  $M/M/1/K$  con un solo servidor, se caracterizan por tener limitado el número de elementos permitidos en cualquier momento en el sistema (se consideran los elementos en la línea de espera y los que están siendo atendidos), dicho número está representado por  $K$ .

En el caso de una estación de producción, la capacidad en la línea de espera se presenta cuando el responsable de ésta decide mantener una cierta cantidad de trabajos frente a la estación (buffer), este material mantiene a la máquina en operación en caso de una falla en el suministro. Además, la tasa de llegadas (salida) debe corregirse ya que en cierta parte del tiempo, el equipo estará bloqueado y no recibirá material.

Las propiedades de los sistemas markovianos con capacidad en la línea (buffer)  $M/M/1/K$  implementadas en este estudio, se pueden consultar en (Shortle et al., 2018). Además, la ecuación *Ley de Little* que permite determinar el valor del *WIP promedio* es expuesta por (Curry and Feldman, 2009).

## 4. Algoritmo híbrido AG-RS propuesto

Las técnicas metaheurísticas híbridas se dividen en dos categorías: la primera consiste en diseñar un solucionador que incluya componentes de una metaheurística en otra, mientras que la segunda combina la metaheurística con otras técnicas típicas de campos como la investigación de operaciones y la inteligencia artificial (Blum et al., 2008). Un representante destacado de la primera categoría es el uso de métodos basados en trayectoria incorporados en métodos basados en población. En el presente estudio se realiza una hibridación de este tipo, donde el método basado en trayectoria es el Recocido Simulado (RS), y el método basado en población es el Algoritmo Genético (AG).

El pseudocódigo del algoritmo híbrido AG-RS propuesto se muestra en la Figura 2. A continuación se explican las características más relevantes de éste.

### 4.1. Esquema inicial

El algoritmo requiere inicialmente de precisar el valor de parámetros que afectarán la búsqueda de la mejor configuración de buffer. El siguiente paso en cualquier Algoritmo Genético tradicional es la creación de una población inicial, para el algoritmo híbrido AG-RS propuesto no es la excepción. En este proceso, la población inicial consta de un número suficiente de configuraciones de buffer que cumplan la restricción 1 del problema BAP3. Cada configuración de buffer se genera aleatoriamente y está compuesta por una cadena codificada de valores enteros que representa las variables de decisión del BAP [ $B = (B_2, B_3, \dots, B_W)$ ].

### 4.2. Evaluación y esquema de selección

Una vez generada la población inicial, cada configuración en la población es evaluada. El *throughput* y el *WIP Promedio* son estimados usando el método de descomposición presentado por (Takahashi et al., 1980) el cual fue desarrollado para analizar sistemas abiertos en red con capacidad restringida, dicho método considera sólo tasas de llegadas y

tiempos de servicio exponenciales. El análisis es realizado con la descomposición de nodo por nodo (o estación por estación) a través de la introducción de una pseudo tasa de llegadas  $\lambda_i^*$  y una tasa efectiva de servicios  $\mu_i^*$ .

```

Comienza
1. Fijar parámetros
   Tamaño de la población,  $P_c, P_m, T_0, \alpha$ , Conservación de la temperatura,
   Número de generaciones, Iteraciones sin mejorar la calidad de la solución
2. Crear población inicial o actual:  $P$ 
3. Evaluar miembros de la población inicial o actual:  $f(B), Q(B)$ ,
4. Definir individuos seleccionables para generar población candidata
5. Crear población candidata:  $P'$ 
   Seleccionar padres para cruce (torneo determinístico y aleatorio)
   Si Aleatorio (0..1) <  $P_c$ , entonces
     Aplicar cruce
   De otro modo añadir una copia de los padres a  $P'$ 
   Si Aleatorio (0..1) <  $P_m$ , entonces
     Seleccionar un hijo
     Aplicar mutación
     Añadir hijos a  $P'$ 
   De otro modo añadir hijos a  $P'$ 
   Si la población  $P'$  está completa, entonces ir a paso 6
   De otro modo, regresar a paso 5
6. Evaluar miembros de la población candidata:  $f(B'), Q(B')$ 
7. Crear nueva población:  $P''$ 
   Aplicar esquema de remplazo, seleccionar soluciones  $B$  y  $B'$ 
   Calcular  $\Delta E = Q(B') - Q(B)$ 
   Si  $\Delta E \leq 0$ , entonces  $B' \rightarrow B''$ 
     En otro caso,
     Si  $\exp(-\Delta E/T) > \text{Aleatorio}(0..1)$ ,
     Entonces  $B' \rightarrow B''$ 
   De otro modo  $B \rightarrow B''$ 
   Si la población  $P''$  está completa, entonces ir a paso 8
   De otro modo, regresar a paso 7
8. Actualizar parámetros y comprobar criterios de paro
    $T \leftarrow \alpha T, P_c \leftarrow P_c', \text{Número de generaciones, Iteraciones sin mejorar la}$ 
    $\text{calidad de la solución}$ 
   Si criterio de paro se cumple, entonces ir a paso 9
   De otro modo, regresar a paso 4
9. Salida  $B'', f(B''), Q(B'')$ 
Fin
    
```

Figura 2: Pseudocódigo del algoritmo híbrido AG-RS.

Definidos los valores del *throughput* y el *WIP Promedio*, éstos se comunican al algoritmo híbrido AG-RS propuesto de una manera iterativa. Aquellas configuraciones que presenten un valor del *throughput* menor al deseado, no se les calculará su *WIP Promedio*, ya que son consideradas como soluciones no factibles.

El esquema de selección sólo considera aquellas configuraciones o soluciones que sean factibles. Antes de iniciar con el procedimiento de cruce, se realiza una selección por elitismo de las 2 mejores soluciones generadas en la población actual y se hace una copia de éstas en la población candidata, de esta manera se garantiza que el proceso de búsqueda nunca dará un paso atrás en cuanto a la calidad de la mejor solución obtenida, sino que un cambio en ésta siempre implicará una mejora. Para el procedimiento de cruce, el padre 1 es seleccionado a través de un torneo determinístico, mientras que el padre 2 es escogido de forma aleatoria, con el fin de incrementar la exploración de nuevas regiones del espacio de búsqueda.

### 4.3. Cruce y mutación

Con el propósito de obtener las mejores configuraciones de buffer, se emplean operadores de cruce y mutación en el algoritmo híbrido AG-RS, similares a los presentados en los Algoritmos Genéticos. Una vez que dos padres han sido seleccionados, se utiliza una probabilidad ( $P_c$ ) para determinar si éstos serán cruzados o no. Los padres que evitan el cruce, automáticamente una copia de ellos pasa a la población candidata.

En el algoritmo híbrido AG-RS la mutación considera sólo aquellos hijos o descendientes generados después del cruce, aquí se elige aleatoriamente a uno de los dos hijos y se emplea una probabilidad ( $P_m$ ) para determinar si éste será mutado o no. Luego de llevar a cabo la mutación, ambos hijos pasan a la población candidata.

En el trabajo desarrollado por (Liu and Tu, 1994) se exhibe un procedimiento de cruce y mutación particular para el BAP, posteriormente (Vergara and Kim, 2009) lo modifican o mejoran; en el presente estudio se utiliza un procedimiento semejante al adoptado por estos últimos autores (ver Figura 3).

```

Cruce: Si  $a$  y  $b$  son los padres (o configuraciones de buffer) seleccionados para el cruce, sus hijos  $a'$  y  $b'$  serán generados a través del siguiente procedimiento:
1.  $a'_i = (a_i + b_i)/2, b'_i = (a_i + b_i)/2$ , para toda  $i$ ;
2. Encontrar todos los elementos (o alelos) que no sean enteros en  $a'$  y  $b'$  (el número total debe ser par) y formar parejas de manera aleatoria como se muestra a continuación:
    $(a'_{i_1}, a'_{j_1}), \dots, (a'_{i_p}, a'_{j_p})$  y  $(b'_{i_1}, b'_{j_1}), \dots, (b'_{i_p}, b'_{j_p})$ ;
3.  $a''_{ik} = a'_{ik} - 0.5$  y  $a''_{jk} = a'_{jk} + 0.5$ , para toda  $k=1, 2, \dots, p$ ;
4.  $b''_{ik} = b'_{ik} + 0.5$  y  $b''_{jk} = b'_{jk} - 0.5$ , para toda  $k=1, 2, \dots, p$ ;
5.  $a'$  y  $b'$  reemplazarán  $a$  y  $b$  en la población candidata.

Mutación: Si el hijo  $a''$  es seleccionado para mutación, la mutación  $a'''$  es obtenida a través del siguiente procedimiento:
1. Seleccionar un par de locaciones  $(i, j)$  de manera aleatoria,
    $2 \leq i \leq M, 2 \leq j \leq M, i \neq j$ ;
2. Para toda  $k \neq i, k \neq j \rightarrow a'''_k = a''_k$ ;
3. Si  $a''_i > 0$  entonces  $a'''_i = a''_i - 1$  y  $a'''_j = a''_j + 1$ 
   Si  $a''_i = 0, a''_j > 0$  entonces  $a'''_i = a''_i + 1$  y  $a'''_j = a''_j - 1$ 
   Si  $a''_i = a''_j = 0$  entonces  $a'''_i = 0$  y  $a'''_j = 0$ 
    
```

Figura 3: Procedimiento de cruce y mutación.

### 4.4. Esquema de remplazo

El esquema de remplazo se basa en el expuesto por (Kose and Kilincci, 2015). La idea clave es determinar cuál configuración de buffer debe permanecer en una población y cuál debe transferirse a la siguiente generación. La prueba de aceptación o rechazo es igual a la presentada en un procedimiento de Recocido Simulado. El algoritmo híbrido AG-RS hace una comparativa entre los integrantes de la población actual y los integrantes de la población candidata. Si el *WIP Promedio* de la configuración candidata  $Q(B')$  es menor al de la configuración actual  $Q(B)$ , entonces la configuración de buffer de la población candidata pasará a la nueva población. De otro modo, la solución candidata no se descarta a la primera, sino que se evalúa la probabilidad de aceptación dada por el *criterio de metrópolis*:

$$P[\text{aceptar } B'] = \begin{cases} 1 & Q(B') < Q(B) \\ \exp\left(-\frac{\Delta E}{T}\right) = \exp\left[-\frac{Q(B') - Q(B)}{T}\right] & Q(B') \geq Q(B) \end{cases} \quad (5)$$

4.5. Criterio de terminación

En los Algoritmos Genéticos, el establecimiento de un número fijo de generaciones es el criterio de terminación más empleado. El algoritmo híbrido AG-RS propuesto, utiliza este criterio para detener su búsqueda de configuraciones de buffer. Sin embargo, el algoritmo también presenta otro criterio de terminación, el cual consiste en establecer un número de iteraciones (o número de generaciones) sin mejorar la calidad de la solución. Finalmente, la mejor configuración de buffer factible que minimice el *WIP Promedio* es seleccionada para resolver el BAP en la línea de producción.

5. Calibración de algoritmo híbrido AG-RS

Se realizaron experimentos estadísticos, con el propósito de calibrar los parámetros del algoritmo híbrido AG-RS y hacer exploraciones eficientes en el espacio de soluciones, en esta sección se muestran los resultados alcanzados. El algoritmo fue programado en lenguaje JAVA, y todos los experimentos fueron desarrollados en una PC con procesador 1.40GHz Intel (R) Celeron (R) 2955U con 4GB de RAM.

El software estadístico utilizado en dichos experimentos fue Minitab 17. La estrategia empleada en la calibración fue basada en una metodología de superficie de respuesta (MSR). Para ello se recurrió al procedimiento mostrado en (Gutiérrez and De la Vara, 2012) el cual considera tres etapas, que son: *cribado*, *búsqueda I o de primer orden* y *búsqueda II o de segundo orden*. A continuación, se describe cada una de éstas.

5.1. Etapa 1: Cribado

Esta etapa tuvo como objetivo identificar aquellos parámetros que tienen mayor influencia en la variable de respuesta, es decir, aquellos parámetros que realmente son significativos en la búsqueda de soluciones del algoritmo híbrido AG-RS.

Se generó un diseño de experimentos Plackett-Burman. La variable de respuesta del experimento fue el *WIP Promedio*. Los ocho parámetros variables del experimento se pueden ver en la Tabla 1, la mayoría de éstos fueron definidos en base a las sugerencias de otros autores, por ejemplo (Goldberg, 1989) recomienda una tasa de cruce alta (0.6-0.90), una tasa de mutación baja (0.01-0.10) y un tamaño de población moderado (50-100), por su parte (Spinellis and Papadopoulos, 2000a) emplean una temperatura inicial=0.5 y una constante de enfriamiento alta (>0.80), finalmente (Kose and Kilincci, 2015) utilizan un número de generaciones bajo (20). La instancia analizada en dicho experimento consideró  $W=5$  máquinas y  $N=20$  buffers disponibles.

Es necesario aclarar que el parámetro de *Iteraciones sin mejorar la calidad de la solución*, define su valor considerando un porcentaje del *Número de generaciones* establecidas en la búsqueda de soluciones.

La gráfica normal y el diagrama de efectos estandarizados (ver Figuras 4 y 5) señalaron que los parámetros: *Tamaño de la población*, *Tasa de cruce (P<sub>c</sub>)*, *Tasa de mutación (P<sub>m</sub>)*, *Constante de enfriamiento (α)* y *Número de generaciones* fueron significativos en el diseño de experimentos realizado. Con estos resultados se finaliza la primera etapa de calibración.

Tabla 1: Parámetros variables.

Parámetros variables	Nivel	
	Bajo	Alto
<b>Algoritmos Genéticos</b>		
Tamaño de la población	50	100
Tasa de cruce (P <sub>c</sub> )	0.60	0.90
Tasa de mutación (P <sub>m</sub> )	0.01	0.10
<b>Recocido Simulado</b>		
Temperatura inicial (T <sub>0</sub> )	0.20	0.50
Constante de enfriamiento (α)	0.80	0.95
Conservación de la temperatura	1	5
<b>Criterio de paro del algoritmo</b>		
Número de generaciones	20	100
Iteraciones sin mejorar la calidad de la solución	(0.5)(NG)	(1)(NG)

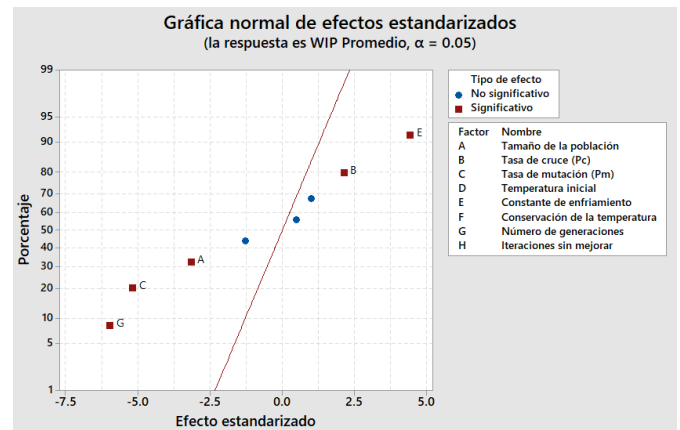


Figura 4: Gráfica normal de efectos estandarizados.

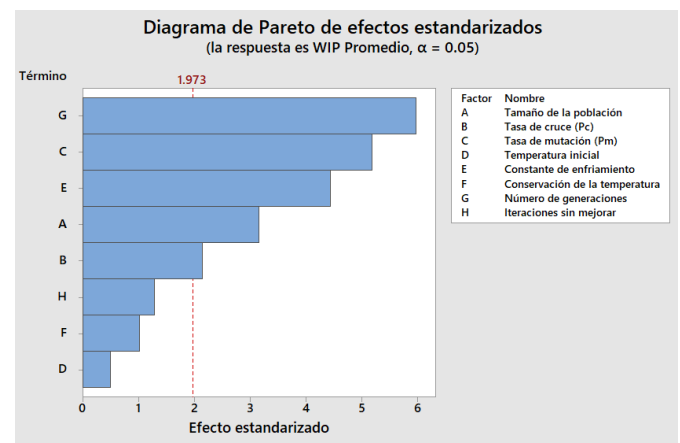


Figura 5: Diagrama de Pareto de efectos estandarizados.

5.2. Etapa 2: Búsqueda I o de primer orden

Esta etapa se aplica cuando se tienen pocos factores ( $k \leq 5$ ), y se sabe que éstos influyen en la variable de respuesta. Aquí se corre un diseño de experimentos de primer orden que permita caracterizar en forma preliminar el tipo de superficie de respuesta y detectar la presencia de curvatura (Gutiérrez and De la Vara, 2012).

Tabla 2: Análisis de la Anova del diseño de experimentos (elementos lineales y curvatura).

<i>Fuente</i>	<i>G.L.</i>	<i>S.C.</i>	<i>M.C.</i>	<i>Valor F</i>	<i>Valor P</i>	<i>Conclusión</i>
Lineal	5	0.010022	0.002004	15.01	0.000	Significativo
Tamaño de la población	1	0.002121	0.002121	15.88	0.000	Significativo
Tasa de cruce ( $P_c$ )	1	0.000063	0.000063	0.47	0.498	
Tasa de mutación ( $P_m$ )	1	0.001894	0.001894	14.18	0.001	Significativo
Constante de enfriamiento ( $\alpha$ )	1	0.000118	0.000118	0.88	0.355	
Número de generaciones	1	0.005826	0.005826	43.62	0.000	Significativo
Curvatura	1	0.001324	0.001324		0.004	Significativo

Tabla 3: Análisis de la Anova del modelo de regresión.

<i>Fuente</i>	<i>G.L.</i>	<i>S.C.</i>	<i>M.C.</i>	<i>Valor F</i>	<i>Valor P</i>	<i>Conclusión</i>
Modelo	3	0.009842	0.003281	12.71	0.000	Significativo
Lineal	3	0.009842	0.003281	12.71	0.000	Significativo
Tamaño de la población	1	0.002121	0.002121	8.22	0.006	Significativo
Tasa de mutación ( $P_m$ )	1	0.001894	0.001894	7.34	0.009	Significativo
Número de generaciones	1	0.005826	0.005826	22.58	0.000	Significativo
Error	60	0.015481	0.000258			
Falta de ajuste	29	0.011341	0.000391	2.93	0.002	Significativo
Error puro	31	0.004140	0.000134			
Total	63	0.025323				

Se desarrolló un diseño de experimentos  $2^5$  con 32 puntos centrales. Con respecto a los parámetros variables, se consideraron sólo aquellos que presentaron una influencia significativa en la variable de respuesta de la etapa 1. La instancia analizada en este experimento consideró  $W=15$  máquinas y  $N=50$  buffers disponibles, es decir, se estudió una instancia de mayor tamaño que la etapa anterior.

Los elementos lineales del análisis de la Anova generado en el diseño de experimentos se muestran en la Tabla 2. El análisis señaló que los parámetros lineales con mayor influencia en la variable de respuesta fueron: *Tamaño de la población*, *Tasa de mutación ( $P_m$ )* y *Número de generaciones*. Además, se detectó que la curvatura en la zona es significativa en la región de experimentación.

El modelo de regresión lineal múltiple generado para el análisis de la superficie de respuesta, que toma en cuenta únicamente a los parámetros significativos es:

$$\hat{y} = 3.3191 - 0.000326x_1 - 0.1710x_2 - 0.000337x_3 \quad (6)$$

Donde,  $x_1$ = Tamaño de la población,  $x_2$ = Tasa de mutación ( $P_m$ ) y  $x_3$ = Número de generaciones. El análisis de Anova de dicho modelo de regresión se muestra en la Tabla 3. Éste señala una falta de ajuste significativa. No es de sorprender esta situación, ya que previamente se detectó la presencia de curvatura en la zona de experimentación.

Por tal motivo, es conveniente ajustar los datos a un modelo de segundo orden. De esta manera, se termina con la etapa 2 de la calibración.

### 5.3. Etapa 3: Búsqueda II o de segundo orden

En el momento en que se detecta la presencia de curvatura, o bien, que la superficie es más complicada que un hiperplano, se corre o se completa un diseño de segundo orden para caracterizar mejor la superficie y modelar la curvatura. Con el modelo ajustado se determinan las condiciones óptimas de operación del proceso (Gutiérrez and De la Vara, 2012).

Se generó un diseño de experimentos de composición central (DCC)  $2^3$  con 6 puntos centrales y 6 puntos axiales. La instancia analizada fue la misma que se estudió en la etapa 2.

El modelo jerárquico de segundo orden de la superficie de respuesta generado es el siguiente:

$$\hat{y} = 3.2430 + 0.000137x_1 + 0.827x_2 + 0.000112x_3 - 1.78x_2^2 - 0.00647x_1x_2 - 0.00311x_2x_3 \quad (7)$$

La Figura 6 muestra las gráficas de contornos de la superficie de respuesta. Del examen de éstas, se determinó el valor de los parámetros que optimizan la superficie de respuesta, es decir, el valor de los parámetros que minimiza el *WIP Promedio*.

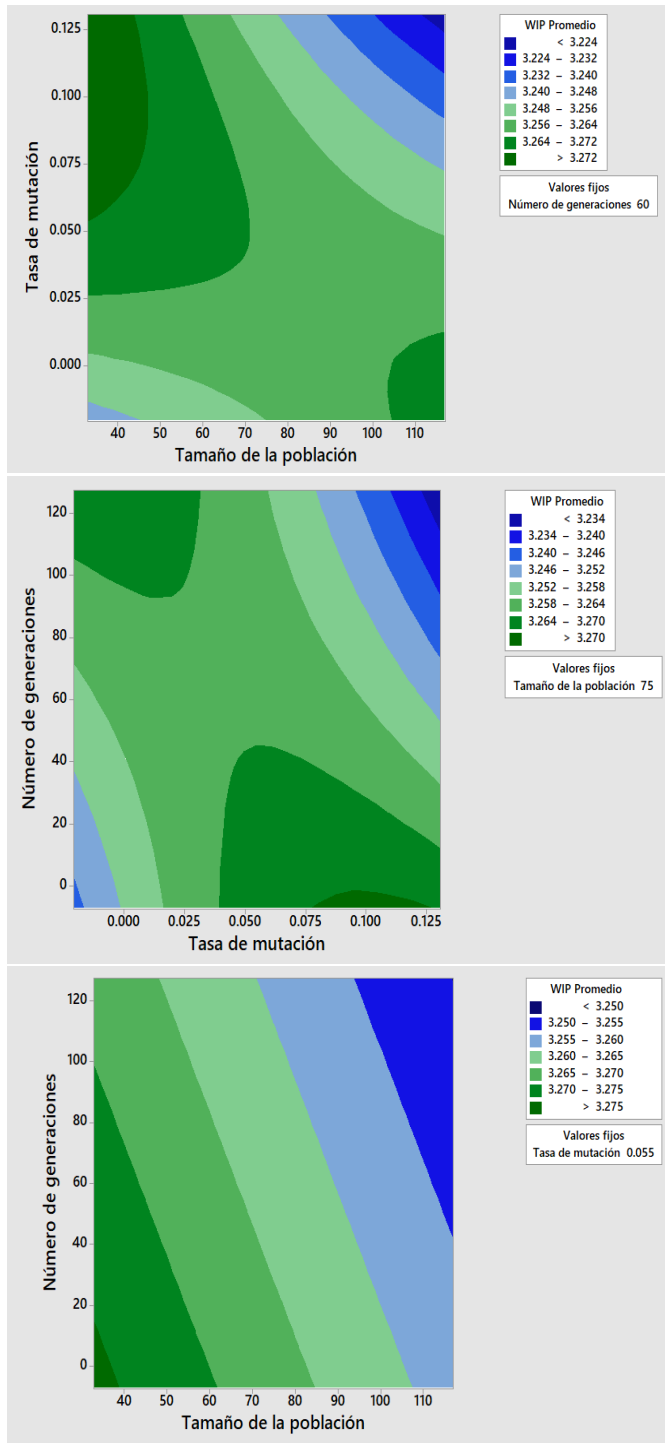


Figura 6: Gráficas de contornos de la superficie de respuesta.

Los valores que optimizan la variable de respuesta son:

- Tamaño de la población: 117
- Tasa de mutación ( $P_m$ ): 0.13
- Número de generaciones: 127

De esta manera, se concluye con el proceso de calibración del algoritmo híbrido AG-RS. La Tabla 4 muestra un resumen del proceso desarrollado y registra los valores calibrados que serán utilizados en los diferentes casos de estudio de la siguiente sección del presente documento.

Tabla 4: Parámetros variables.

Parámetros	Etapa			Valor calibrado
	1	2	3	
<b>Algoritmos Genéticos</b>				
Tamaño de la población			117	<b>117</b>
Tasa de cruce ( $P_c$ )		0.60		<b>0.60</b>
Tasa de mutación ( $P_m$ )			0.13	<b>0.13</b>
<b>Recocido Simulado</b>				
Temperatura inicial ( $T_0$ )	0.50			<b>0.50</b>
Constante de enfriamiento ( $\alpha$ )		0.80		<b>0.80</b>
Conservación de la temperatura	1			<b>1</b>
<b>Criterio de paro del algoritmo</b>				
Número de generaciones			127	<b>127</b>
Iteraciones sin mejorar la calidad de la solución	1			<b>(1)(NG)=127</b>

### 6. Resultados numéricos del algoritmo híbrido AG-RS

En esta sección, se reportan las mejores soluciones alcanzadas por el algoritmo híbrido AG-RS. Se investigan líneas cortas, medianas, largas y muy largas. Considerando que éstas sean líneas balanceadas, y en algunos casos, no balanceadas. Las líneas no balanceadas se limitan al caso de  $\mu$ -No balanceada. Además, todas las instancias estudiadas tienen la característica de ser líneas de producción confiables.

Se recuerda que para resolver el modelo matemático BAP3, es necesario establecer el número de buffers disponibles ( $N$ ) y definir el nivel de throughput deseado ( $f^*$ ) en la línea de producción. La tasa de llegadas ( $\lambda$ ), las tasas de servicios ( $\mu$ ) de las máquinas y el throughput están dados en piezas/min, mientras que el WIP Promedio se mide en piezas.

El análisis está dividido en seis casos de estudio, las características de éstos están basadas en los presentados por (Kose and Kilincci, 2015), con excepción del caso 5 que es una propuesta de este estudio.

- Caso 1: Líneas balanceadas con 5, 10 y 15 máquinas en serie (ver Tablas 5-7).
- Caso 2: Líneas no balanceadas con 8 máquinas en serie y un cuello de botella (ver Tablas 8-10).
- Casos 3 y 4: Líneas no balanceadas con 8 máquinas en serie y dos cuellos de botella (ver Tablas 11-16).
- Caso 5: Líneas balanceadas largas con 20, 30 y 40 máquinas (ver Tabla 17).
- Caso 6: Líneas balanceadas muy largas con 50, 60, 80 y 100 máquinas (ver Tabla 18).

Tabla 5: Caso 1: a) Resultados para una línea balanceada con  $W=5$  máquinas,  $\lambda=1$  y  $\mu=2$ .

$N$	$f^*$	$B^*$	Throughput $f(B^*)$	WIP Promedio $Q(B^*)$
10	0.82	1, 2, 2, 5	0.8221	3.0894
15	0.90	1, 5, 3, 6	0.9011	3.5803
20	0.95	2, 4, 7, 7	0.9503	3.9689
30	0.98	3, 6, 1, 11	0.9801	4.2564

Tabla 6: Caso 1: b) Resultados para una línea balanceada con W=10 máquinas, λ=1 y μ=2.

N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
5	0.20	0, 0, 0, 0, 0, 0, 0, 1, 4	0.2075	1.2640
10	0.35	0, 0, 0, 1, 1, 1, 1, 1, 5	0.3521	2.1851
15	0.50	0, 0, 2, 1, 2, 2, 2, 3, 3	0.5000	3.2661
20	0.60	0, 1, 1, 2, 2, 2, 2, 6, 2, 4	0.6001	4.1227
30	0.78	1, 2, 2, 3, 3, 4, 4, 4, 7	0.7803	6.0701
50	0.90	1, 4, 5, 4, 6, 8, 6, 8, 8	0.9001	7.6485

Tabla 7: Caso 1: c) Resultados para una línea balanceada con W=15 máquinas, λ=1 y μ=2.

N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
5	0.10	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5	0.1243	1.0930
10	0.15	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 7	0.1530	1.3460
15	0.25	0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2, 5	0.2501	2.2377
20	0.35	0, 0, 0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 4	0.3520	3.2501
30	0.48	0, 0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 5	0.4817	4.7058
50	0.65	0, 1, 2, 2, 3, 3, 2, 2, 4, 5, 6, 5, 5, 3, 9	0.6502	6.9986

Tabla 8: Caso 2: a) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y un cuello de botella μ=1.9.

Cuello de botella	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
W <sub>2</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2371	1.1891
	8	0.45	0, 1, 0, 1, 1, 2, 3	0.4513	2.3705
W <sub>3</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2362	1.1856
	8	0.45	0, 0, 1, 1, 1, 2, 3	0.4582	2.3648
W <sub>4</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2359	1.1835
	8	0.45	0, 1, 0, 2, 1, 1, 3	0.4513	2.3814

Tabla 9: Caso 2: b) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y un cuello de botella μ=1.5.

Cuello de botella	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
W <sub>2</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2329	1.2153
	8	0.45	0, 0, 2, 1, 1, 1, 3	0.4503	2.4236
W <sub>3</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2275	1.1940
	8	0.45	0, 1, 2, 0, 1, 2, 2	0.4504	2.5376
W <sub>4</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2253	1.1812
	8	0.45	0, 0, 1, 1, 2, 2, 2	0.4567	2.4697

Tabla 10: Caso 2: c) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y un cuello de botella μ=1.

Cuello de botella	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
W <sub>2</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2225	1.2733
	8	0.45	0, 1, 1, 1, 1, 1, 3	0.4677	2.7824
W <sub>3</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2081	1.2136
	8	0.45	0, 1, 1, 1, 1, 3, 1	0.4510	2.8171
W <sub>4</sub>	2	0.20	0, 0, 0, 0, 0, 0, 0, 2	0.2026	1.1778
	8	0.45	0, 1, 1, 1, 1, 2, 2	0.4587	2.8980

Tabla 11: Caso 3: a) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y dos cuellos de botella [W<sub>2</sub>, W<sub>6</sub>]; μ(W<sub>6</sub>)=1.9.

μ(W <sub>2</sub> )	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
1.9	8	0.45	0, 0, 1, 1, 1, 2, 3	0.4591	2.3875
1.5	8	0.45	0, 1, 0, 1, 2, 2, 2	0.4509	2.4761
1	8	0.40	0, 1, 1, 0, 1, 2, 3	0.4023	2.3944

Tabla 12: Caso 3: b) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y dos cuellos de botella [W<sub>2</sub>, W<sub>6</sub>]; μ(W<sub>6</sub>)=1.5.

μ(W <sub>2</sub> )	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
1.9	8	0.45	0, 0, 1, 3, 1, 2, 1	0.4508	2.4693
1.5	8	0.45	0, 0, 1, 1, 2, 2, 2	0.4536	2.5661
1	8	0.40	0, 0, 1, 2, 1, 1, 3	0.4012	2.4490

Tabla 13: Caso 3: c) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y dos cuellos de botella [W<sub>2</sub>, W<sub>6</sub>]; μ(W<sub>6</sub>)=1.

μ(W <sub>2</sub> )	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
1.9	8	0.42	0, 0, 2, 1, 1, 2, 2	0.4232	2.5869
1.5	8	0.42	0, 0, 1, 1, 2, 1, 3	0.4226	2.6804
1	8	0.40	0, 0, 1, 1, 2, 2, 2	0.4026	2.7263

Tabla 14: Caso 4: a) Resultados para una línea no balanceada con W=8 máquinas, λ=1, μ=2 y dos cuellos de botella [W<sub>2</sub>, W<sub>4</sub>]; μ(W<sub>4</sub>)=1.9.

μ(W <sub>2</sub> )	N	f*	B*	Throughput f(B*)	WIP Promedio Q(B*)
1.9	8	0.45	0, 0, 1, 1, 1, 2, 3	0.4584	2.3840
1.5	8	0.45	0, 0, 2, 1, 3, 1, 1	0.4507	2.4561
1	8	0.40	0, 1, 0, 1, 1, 2, 3	0.4025	2.3840



Tabla 15: Caso 4: b) Resultados para una línea no balanceada con W=8 máquinas,  $\lambda=1$ ,  $\mu=2$  y dos cuellos de botella [ $W_2, W_4$ ];  $\mu(W_4)=1.5$ .

$\mu(W_2)$	N	$f^*$	$B^*$	Throughput $f(B^*)$	WIP Promedio $Q(B^*)$
1.9	8	0.45	0, 0, 1, 1, 2, 2, 2	0.4541	2.4714
1.5	8	0.45	0, 0, 2, 1, 1, 2, 2	0.4543	2.5883
1	8	0.40	0, 0, 1, 1, 2, 2, 2	0.4061	2.4768

Tabla 16: Caso 4: c) Resultados para una línea no balanceada con W=8 máquinas,  $\lambda=1$ ,  $\mu=2$  y dos cuellos de botella [ $W_2, W_4$ ];  $\mu(W_4)=1$ .

$\mu(W_2)$	N	$f^*$	$B^*$	Throughput $f(B^*)$	WIP Promedio $Q(B^*)$
1.9	8	0.42	0, 0, 2, 1, 1, 1, 3	0.4263	2.6520
1.5	8	0.42	0, 0, 2, 1, 1, 2, 2	0.4263	2.7273
1	8	0.40	0, 0, 3, 1, 1, 1, 2	0.4014	2.7849

Tabla 17: Caso 5: Resultados para líneas balanceadas largas con  $\lambda=1$ ,  $\mu=10$ .

W	N	$f^*$	$B^*$	Throughput $f(B^*)$	WIP Promedio $Q(B^*)$
20	60	0.95	0, 1, 1, 5, 1, 2, 2, 2, 2, 2, 7, 1, 2, 2, 2, 5, 8, 2, 13	0.9501	2.0861
30	90	0.85	0, 1, 6, 2, 4, 2, 1, 1, 0, 1, 1, 7, 3, 6, 7, 2, 4, 1, 4, 1, 3, 5, 2, 2, 4, 2, 2, 2, 14	0.8500	2.7856
40	120	0.75	2, 0, 0, 2, 3, 1, 3, 2, 1, 2, 5, 2, 2, 2, 2, 13, 3, 3, 5, 2, 4, 3, 2, 4, 2, 3, 3, 2, 2, 2, 3, 5, 2, 6, 0, 2, 3, 3, 14	0.7500	3.2598

Tabla 18: Caso 6: Resultados para líneas balanceadas muy largas con  $\lambda=1$ ,  $\mu=10$ .

W	N	$f^*$	$B^*$	Throughput $f(B^*)$	WIP Promedio $Q(B^*)$
50	150	0.65	2, 1, 6, 2, 2, 0, 2, 2, 0, 2, 3, 3, 2, 3, 2, 0, 1, 5, 2, 3, 0, 3, 4, 2, 5, 3, 0, 2, 3, 3, 2, 3, 4, 3, 2, 2, 3, 1, 2, 5, 16, 1, 3, 3, 3, 3, 6, 3, 17	0.6500	3.5070
60	180	0.65	1, 2, 3, 1, 1, 0, 1, 2, 3, 5, 1, 1, 1, 2, 4, 4, 0, 2, 2, 1, 1, 5, 3, 8, 1, 1, 1, 5, 7, 3, 3, 5, 6, 1, 2, 2, 3, 1, 0, 0, 4, 4, 4, 2, 2, 3, 2, 1, 1, 3, 3, 3, 1, 2, 3, 4, 2, 5, 36	0.6500	4.2165
80	240	0.55	5, 0, 2, 3, 0, 1, 3, 1, 7, 0, 4, 3, 5, 4, 3, 0, 2, 6, 3, 4, 3, 3, 4, 4, 1, 4, 1, 1, 2, 1, 0, 1, 2, 2, 1, 3, 2, 1, 5, 1, 2, 5, 3, 0, 2, 5, 1, 1, 2, 5, 3, 2, 1, 1, 3, 6, 2, 0, 3, 6, 2, 1, 2, 4, 1, 2, 1, 3, 2, 2, 4, 3, 3, 1, 3, 5, 2, 3, 45	0.5500	4.7040
100	300	0.45	2, 6, 0, 3, 1, 1, 6, 1, 0, 5, 1, 5, 0, 1, 1, 2, 1, 2, 5, 1, 2, 2, 3, 4, 2, 6, 0, 2, 0, 1, 3, 1, 1, 2, 2, 1, 2, 2, 2, 3, 2, 5, 3, 2, 4, 3, 3, 6, 3, 3, 6, 0, 4, 3, 2, 4, 7, 5, 2, 4, 3, 1, 6, 1, 3, 2, 0, 2, 2, 6, 2, 1, 4, 0, 2, 3, 2, 4, 4, 3, 4, 2, 5, 1, 3, 1, 2, 0, 2, 0, 3, 3, 3, 3, 0, 4, 4, 5, 48	0.4500	4.7999

Los resultados reportados en el caso 1 (ver Tablas 5-7), señalan una clara tendencia por asignar una mayor cantidad de buffers en las máquinas que se encuentran hacia el final de la línea de producción.

Con respecto a los casos 2, 3 y 4 (ver Tablas 8-16), a diferencia de (Kose and Kilincci, 2015) en donde se registra una mayor asignación de buffers en las máquinas consideradas como cuellos de botella, el algoritmo híbrido AG-RS presenta configuraciones de buffer similares a las mostradas en el caso 1, es decir, un mayor número de buffers colocados en las últimas máquinas del proceso.

Finalmente, en los casos 5 y 6 (ver Tablas 17 y 18) las configuraciones exhiben una asignación variada, sin embargo en todas las instancias analizadas, la última máquina acumula la mayor cantidad de buffers. También es posible observar que cuanto mayor sea el tamaño de la línea de producción, es más complicado alcanzar una buena *tasa promedio de producción* o *throughput*.

## 7. Estudio comparativo de la eficiencia computacional del algoritmo híbrido AG-RS

En los siguientes apartados, se expone un estudio que compara la eficiencia computacional del algoritmo híbrido AG-RS con respecto a un RS y AG estándar, los cuales emplean *métodos de evaluación de: simulación y analíticos*. En dicho estudio, se considera también al algoritmo híbrido GAA desarrollado por (Kose and Kilincci, 2015).

### 7.1. AG-RS Vs RS/AG/GAA con método de evaluación basado en simulación

Antes de hacer el análisis de la eficiencia computacional, es importante conocer las diferencias entre los algoritmos híbridos AG-RS y GAA. La Tabla 19 presenta una comparativa de las características más importantes de ambos.

Tabla 19: Características más importantes de los algoritmos híbridos AG-RS y GAA.

Característica	GAA	AG-RS
Problema del BAP	BAP1	BAP3
Método de optimización	Algoritmos genéticos-Recocido simulado	Algoritmos genéticos-Recocido simulado
Método de evaluación	Simulación	Método analítico de descomposición
Lenguaje de programación	Matlab	Java
<b>Algoritmos Genéticos</b>		
Tamaño de la población	50	117
Tasa de cruce ( $P_c$ )	0.80	0.60
Tasa de mutación ( $P_m$ )	0.033	0.13
<b>Recocido Simulado</b>		
Temperatura inicial ( $T_0$ )	0.50	0.50
Constante de enfriamiento ( $\alpha$ )	0.90	0.80
<b>Criterio de paro del algoritmo</b>		
Número de generaciones	20	127

Son claras las diferencias que existen entre uno y otro, en estructura y objetivos de solución. Un aspecto importante por señalar es el método de evaluación usado. El algoritmo híbrido AG-RS emplea un método analítico de descomposición, según (Demir *et al.*, 2014) la principal ventaja de este tipo de métodos es su eficiencia computacional y su precisión para alcanzar una solución.

La Tabla 20 muestra los tiempos obtenidos por ambos algoritmos considerando los primeros cuatro casos de estudio de la sección anterior, además se incorporan los tiempos conseguidos por los algoritmos RS y GA estándar reportados en el mismo trabajo de (Kose and Kilincci, 2015), éstos implementan un método de evaluación basado en simulación. A su vez, la Tabla 21 establece una comparativa de los tiempos para instancias de líneas muy largas.

Tabla 20: Tiempos promedios de solución CPU (segundos) de los casos 1-4.

Método	Tiempo promedio de solución por caso de estudio (segundos)				Tiempo promedio final
	Caso 1	Caso 2	Caso 3	Caso 4	
RS	36,806	21,601	21,806	23,545	25,939.5
AG	21,157	18,557	17,431	20,211	19,339.0
GAA	19,723	16,069	16,057	17,023	17,218.0
AG-RS	<b>0.539</b>	<b>0.450</b>	<b>0.445</b>	<b>0.419</b>	<b>0.463</b>

Tabla 21: Tiempos promedios de solución CPU (segundos) en líneas muy largas.

W	N	Tiempo promedio de solución (segundos)			
		RS	AG	GAA	AG-RS
50	150	73,462.6	69,881.3	69,237.7	<b>2.478</b>
60	180	115,572.6	113,290.5	112,860.6	<b>3.487</b>
80	240	241,841.0	220,639.5	220,040.0	<b>4.113</b>
100	300	157,948.4	142,207.7	141,671.3	<b>6.183</b>

Resulta evidente la ventaja de utilizar métodos de evaluación analíticos con respecto a métodos basados en simulación. Los tiempos obtenidos por el algoritmo híbrido AG-RS están muy por debajo a un RS o AG estándar, incluso la eficiencia computacional de éste es muy superior al algoritmo híbrido GAA en todos los casos e instancias analizadas.

### 7.2. AG-RS Vs RS/AG con método de evaluación analítico

En este apartado, se hace una comparativa del algoritmo híbrido AG-RS con respecto a otros algoritmos RS y AG estándar, cuyo método de evaluación resulta ser analítico.

Los algoritmos RS y GA utilizados en esta comparativa fueron los presentados en (Spinellis and Papadopoulos, 2000a) y (Spinellis and Papadopoulos, 2000b) respectivamente, los cuales resuelven el modelo matemático BAP1. Es importante aclarar que el software de ejecución de éstos, nombrado "Production Line Calculator Wizard", fue obtenido de (Papadopoulos *et al.*, 2009). Además, DECO-1 se empleó como método de evaluación (para mayor información sobre éste, se recomienda consultar a las fuentes citadas).

El número de iteraciones del software mencionado en el párrafo anterior, fue realizado en la misma computadora donde se desarrollaron todos los experimentos con el algoritmo híbrido AG-RS, las características de dicha computadora fueron descritas previamente en la sección 5.

La Tabla 22 exhibe los tiempos registrados (en segundos) por cada uno de los algoritmos, en diferentes instancias analizadas de líneas balanceadas.

Tabla 22: Tiempos promedios de solución CPU (segundos) en líneas balanceadas.

Líneas	W	N	Tiempo promedio de solución (segundos)		
			RS	AG	AG-RS
Cortas y medianas	5	15	<1	<1	<1
	10	30	<1	<1	<1
	15	45	1	<1	<1
Largas	20	60	2	1	<1
	30	90	5	2	<b>1.366</b>
	40	120	9	3	<b>2.072</b>
Muy largas	50	150	13	4	<b>2.478</b>
	60	180	18	6	<b>3.487</b>
	80	240	37	12	<b>4.113</b>
	100	300	61	22	<b>6.183</b>

Los resultados demuestran que en instancias de líneas cortas y medianas (con 5, 10 y 15 máquinas), los algoritmos presentan tiempos de ejecución muy similares. Sin embargo, en instancias de líneas largas (con 20, 30 y 40 máquinas) el algoritmo RS estándar es el que mayor tiempo de ejecución requiere, a su vez, los algoritmos AG y AG-RS presentan tiempos muy parecidos, aunque el algoritmo híbrido resulta ser ligeramente más eficiente en líneas de 30 y 40 máquinas. Finalmente, en instancias de líneas muy largas (con 50, 60, 80 y 100 máquinas) el algoritmo híbrido AG-RS muestra una eficiencia computacional superior a los algoritmos RS y AG

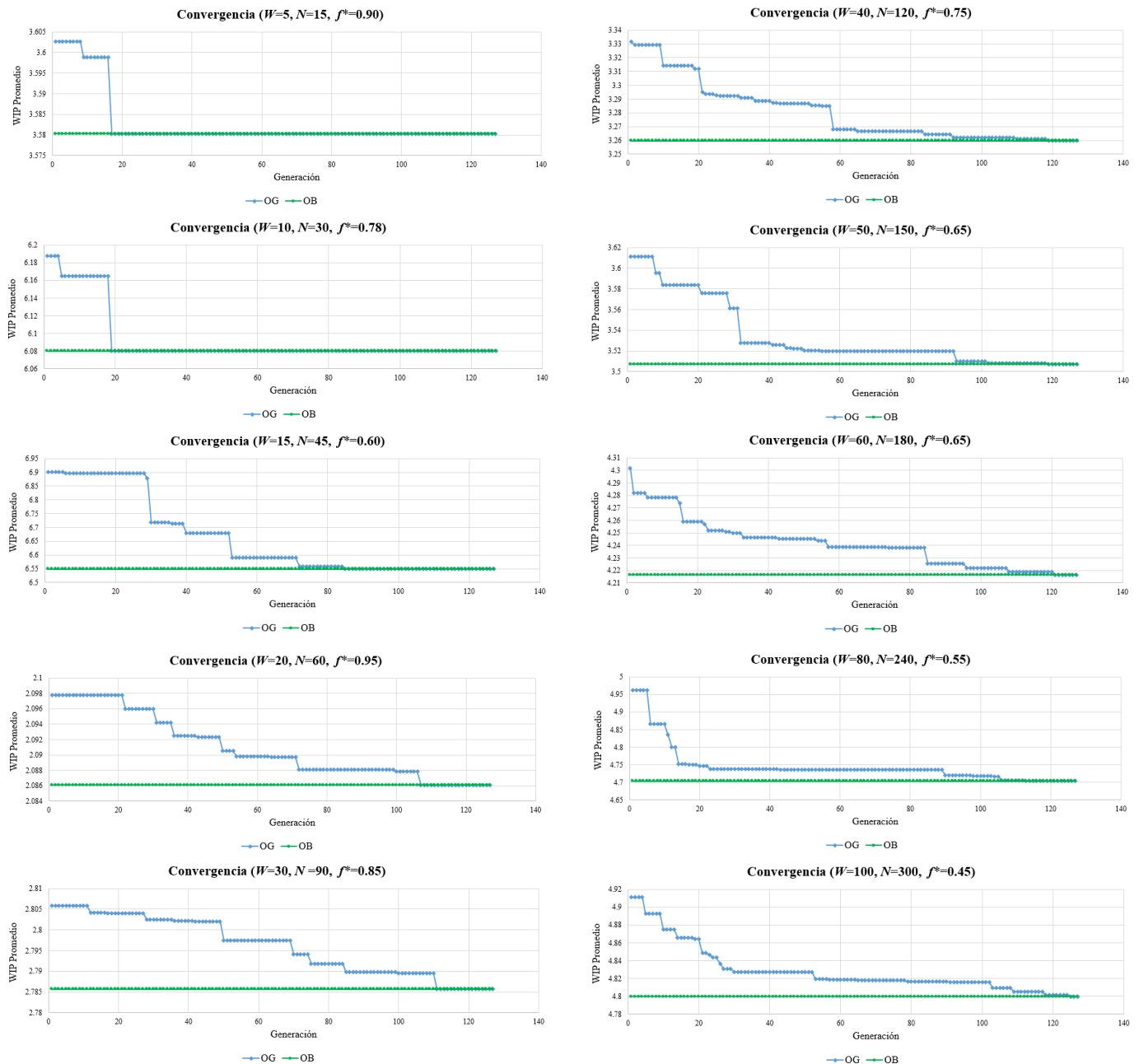


Figura 7: Gráficas de convergencia del algoritmo híbrido AG-RS.

estándar, por lo que pudiera concluirse que éste mejora su rendimiento con respecto a procedimientos estándar a medida que aumenta el tamaño de las instancias analizadas.

### 8. Convergencia del algoritmo híbrido AG-RS

La velocidad de convergencia del algoritmo híbrido AG-RS varía según el tamaño de la instancia analizada. La Figura 7 muestra la manera en que converge la mejor solución encontrada en cada generación, o bien, el óptimo de cada generación (OG) con respecto a la mejor solución u óptimo de la búsqueda (OB). Se puede observar que en *líneas cortas*, la convergencia se obtiene con menos de 20 generaciones. Para *líneas largas y muy largas*, el algoritmo requiere de un mayor número de éstas (aproximadamente 100) para converger de forma exitosa.

### 9. Conclusiones

En el presente estudio, se propuso el algoritmo híbrido metaheurístico AG-RS para resolver el BAP bajo el enfoque de minimizar el promedio de inventario en proceso. Un aspecto a resaltar, es la explicación del proceso empleado en la calibración de éste, el cual estuvo basado en la metodología de superficie de respuesta (MSR). Se resolvieron diferentes instancias de *líneas cortas, medianas, largas y muy largas* abiertas en serie. Luego se realizaron pruebas comparativas para determinar la eficiencia computacional de dicho algoritmo híbrido con respecto a otros procedimientos reportados en la literatura.

Los primeros resultados obtenidos en el estudio comparativo permiten concluir que el algoritmo híbrido AG-RS es notablemente más eficiente que los algoritmos RS y AG

estándar, los cuales emplean métodos de evaluación basados en simulación, incluso supera en demasía a su antecesor el algoritmo híbrido GAA.

Posteriormente, el algoritmo híbrido AG-RS fue probado contra otros algoritmos metaheurísticos estándar cuyo proceso de evaluación lo realizan a través de un método analítico. Los resultados generados demuestran que el algoritmo híbrido propuesto comienza a mejorar su rendimiento al momento de analizar instancias de mayor tamaño, sobre todo, es clara su superioridad al analizar *líneas muy largas*. Por lo anterior, se puede concluir que la potencia del algoritmo híbrido AG-RS presentado en este trabajo resulta ser mejor que otros algoritmos RS y AG estándar.

Finalmente, en futuros trabajos se recomienda analizar la calidad de las soluciones alcanzadas por el algoritmo híbrido AG-RS. Actualmente, no se han reportado estudios que analicen el modelo matemático *BAP3* con el método de descomposición presentado en esta investigación, por lo que resulta complicado afirmar si se alcanza el óptimo, o bien, que tan lejos o tan cerca se está de él. Además, sería atractivo investigar líneas de producción que presentan fallas en sus máquinas (líneas poco confiables), o incluso, resolver casos de estudio aplicados en procesos de manufactura reales.

## Agradecimientos

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el financiamiento de esta investigación con número de registro CVU: 375571. Un reconocimiento a Juana Cinthia Lizbeth Nava Torres, Rafael Paniagua Soto y Juan Pablo Gallardo Ochoa por su apoyo en la fase de programación.

## Referencias

- Amiri, M., & Mohtashami, A. (2012). Buffer allocation in unreliable production lines based on design of experiments, simulation, and genetic algorithm. *International Journal of Advanced Manufacturing Technology*, 62, 371–383.
- Ariyani, A. K., Mahmudy, W. F., & Anggodo, Y. P. (2018). Hybrid genetic algorithms and simulated annealing for multi-trip vehicle routing problem with time windows. *International Journal of Electrical and Computer Engineering*, 8(6), 4713–4723.
- Blum, C., Blesa Aguilera, M. J., Roli, A., & Sampels, M. (2008). *Hybrid metaheuristics an emerging approach to optimization*. Springer, Berlin.
- Costa, A., Alfieri, A., Matta, A., & Fichera, S. (2015). A parallel tabu search for solving the primal buffer allocation problem in serial production systems. *Computers & Operations Research*, 97–112.
- Cruz, F. R., Kendall, G., While, L., Duarte, A. R., & Brito, N. L. (2012). Throughput maximization of queueing networks with simultaneous minimization of servicer rates and buffers. *Mathematical Problems in Engineering*, 1-19. doi:10.1155/2012/692593
- Curry, G., & Feldman, R. (2009). *Manufacturing Systems Modeling and Analysis*. Springer, Berlin.
- Demir, L., Tunali, S., & Tursel Eliyi, D. (2014). The state of the art on buffer allocation problem: a comprehensive survey. *Journal of Intelligent Manufacturing*, 25(3), 371–392.
- Demir, L., & Tunali, S. (2008). A new approach for optimal buffer allocation in unreliable production lines. *Proceedings of 38th International Conference on Computers*, (págs. 1962-1970).
- Dolgui, A., Ereemeev, A. V., & Sigaev, V. S. (2007). HBBA: hybrid algorithm for buffer allocation in tandem production lines. *Journal of Intelligent Manufacturing*, 18, 411–420.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (Primera ed.). Addison-Wesley Professional, United States of America.
- Gutiérrez Pulido, H., & De la Vara Salazar, R. (2012). *Análisis y diseño de experimentos* (Tercera ed.). McGraw-Hill, México.
- Huilcapi, V., Lima, B., Blasco, X., & Herrero, J. M. (2018). Multi-objective optimization in modeling and control for rotary inverted pendulum. *Revista Iberoamericana de Automática e Informática Industrial*, 15(4), 363–373.
- Kose, S. Y., & Kilincci, O. (2015). Hybrid approach for buffer allocation in open serial production lines. *Computers & Operations Research*, 60, 67–78.
- Kose, S. Y., & Kilincci, O. (2018). A multi-objective hybrid evolutionary approach for buffer allocation in open serial production lines. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-018-1435-6
- Liu, C., & Tu, F. S. (1994). Buffer allocation via the genetic algorithm. In: *Proceedings of 33rd conference on decision and control*. 609–610.
- Mohtashami, A. (2014). A new hybrid method for buffer sizing and machine allocation in unreliable production and assembly lines with general distribution time-dependent parameters. *International Journal of Advanced Manufacturing Technology*, 74, 1577–1593.
- Nahas, N., & Nourelfath, M. (2018). Joint optimization of maintenance, buffers and machines in manufacturing lines. *Engineering Optimization*, 50(1), 37–54.
- Nahas, N., Nourelfath, M., & Ait-Kadi, D. (2009). Selecting machines and buffers in unreliable series-parallel production lines. *International Journal of Production Research*, 47(14), 3741–3774.
- Nahas, N., Nourelfath, M., & Gendreau, M. (2014). Selecting machines and buffers in unreliable assembly/disassembly manufacturing networks. *International Journal of Production Economics*, 154, 113–126.
- Narasimhamu, K. L., Reddy, V. V., & Rao, C. (2014). Optimal buffer allocation in tandem closed queueing network with single server using PSO. *Procedia Materials Science*, 5, 2084–2089.
- Narasimhamu, K. L., Reddy, V. V., & Rao, C. (2015). Optimization of buffer allocation in manufacturing system using particle swarm optimization. *International Review on Modelling and Simulations*, 8(2).
- Ortiz-Quisbert, M. E., Duarte-Mermoud, M. A., Milla, F., & Castro-Linares, R. (2016). Fractional adaptive control optimized by genetic algorithms, applied to automatic voltage regulators. *Revista Iberoamericana de Automática e Informática industrial*, 13(4), 403–409.
- Papadopoulos, C. T., O’Kelly, M. J., Vidalis, M. J., & Spinellis, D. (2009). *Analysis and design of discrete part production lines*. New York: Springer.
- Papadopoulos, H. T., & Vidalis, M. I. (2001). Minimizing WIP inventory in reliable production lines. *International Journal of Production Economics*, 70, 185–197.
- Rodríguez-Blanco, T., Sarabia, D., & De Prada, C. (2018). Real-time optimization using the modifier adaptation methodology. *Revista Iberoamericana de Automática e Informática industrial*, 15(2), 133–144.
- Shi, L., & Men, S. (2003). Optimal buffer allocation in production lines. *IIE Transactions*, 35, 1–10.
- Shortle, J., Thompson, J., Gross, D., & Harris, C. (2018). *Fundamentals of Queueing Theory* (Fifth ed.). Wiley, United States of America.
- Spinellis, D. D., & Papadopoulos, C. T. (2000a). A simulated annealing approach for buffer allocation in reliable production lines. *Annals of Operations Research*, 93, 373–384.
- Spinellis, D. D., & Papadopoulos, C. T. (2000b). Stochastic algorithms for buffer allocation in reliable production lines. *Mathematical Problems in Engineering*, 5, 441–458.
- Spinellis, D., Papadopoulos, C., & Smith, J. M. (2000). Large production line optimisation using simulated annealing. *International Journal of Production Research*, 38(3), 509–541.
- Su, C., Shi, Y., & Dou, J. (2017). Multi-objective optimization of buffer allocation for remanufacturing system based on TS-NSGAI hybrid algorithm. *Journal of Cleaner Production*, 166, 756–770.
- Takahashi, Y., Miyahara, H., & Hasegawa, T. (1980). An approximation method for open restricted queueing networks. *Operations Research*, 28(3), 594–602.
- Vergara, H. A., & Kim, D. S. (2009). A new method for the placement of buffers in serial production lines. *International Journal of Production Research*, 47(16), 4437–4445.
- Wei, H., Li, S., Jiang, H., Hu, J., & Hu, J. (2018). Hybrid genetic simulated annealing algorithm for improved flow shop scheduling with makespan criterion. *Applied Sciences*, 8(2621), 1–20.
- Weiss, S., Schwarz, J. A., & Stollet, R. (2018). The buffer allocation problem in production lines: Formulations, solution methods, and instances. *IIE Transactions*. doi:10.1080/24725854.2018.1442031