

**XIX SEMANA DA  
ENGENHARIA  
DE PRODUÇÃO  
E MECÂNICA  
SULAMERICANA**

**XIX SEPROSUL**

**ANAIS**





# **ANAIS** DA XIX SEMANA DA ENGENHARIA DE PRODUÇÃO E MECÂNICA SULAMERICANA

## **ORGANIZADORES**

*Carlos Alberto Bavastri*  
*Marcio Brandani Tenório*  
*Natália Gaspar Lamb*

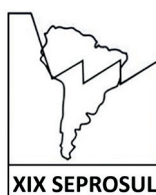
**1ª edição**

Curitiba, 11 a 13 de novembro de 2019



# ANAIS DO XIX SEPROSUL

*Realização:*



*Apoio:*





## COMITÊ CIENTÍFICO

Adrian Toncovich  
Alessandra de Paula  
Ângela de Moura Ferreira Danilevicz  
Arinei Carlos Lindbeck da Silva  
Carlos Alberto Bavastri  
Cassia Marie Lie Ugaya  
Claudia Zarate  
Eduardo Afonso Ribeiro  
Eduardo Marcio de Oliveira Lopes  
Fabiano Oscar Drozda  
Izabel Cristina Zattar  
Jucélio Tomás Pereira  
Julio Cezar Mairesse Siluk  
Luciano Brito Rodrigues  
Marcell Mariano Correa Maceno  
Marcos Augusto Mendes Marques  
Maria Lucia Okimoto  
Mariano Febbo  
Monica Beltrami  
Myriam Eugênia Ramalho Prata Barbejat  
Nancy Alves  
Oscar Daniel Quiroga  
Robson Seleme

## COMISSÃO ORGANIZADORA

**PRESIDENTE:** Carlos Alberto Bavastri

**VICE-PRESIDENTE:** Robson Seleme

### MEMBROS:

Marcio Tenorio Brandani  
Izabel Cristina Zattar  
Ângela de Moura Ferreira Danilevicz  
Julio Cezar Mairesse Siluk  
Myriam Eugênia Ramalho Prata Barbejat  
Guillermo Rodríguez  
Alejandro Gutierrez  
Adrian Andres Toncovich  
Oscar Daniel Quiroga  
Cassia Marie Lie Ugaya  
Maria Lucia Leite Ribeiro Okimoto  
Alejandro Mateos



Diante do movimento de integração da América do Sul, existe a necessidade de fortalecer a integração acadêmica nos países do Cone Sul. Assim, a **SEPROSUL** tem sido realizada há vários anos para possibilitar a troca de experiências e estabelecimento de estratégias comuns para o desenvolvimento da pesquisa e ensino na área de Engenharia Industrial na América do Sul. O **SEPROSUL** foi criado pelo Núcleo Disciplinar de Engenharia Mecânica e da Produção (NDIMP) que por sua vez está inserido na Associação Universidades Grupo Montevideo (AUGM).

A AUGM é uma Rede de Universidades públicas, autônomas e autogovernadas da Argentina, Bolívia, Brasil, Chile, Paraguai e Uruguai que, em função de suas semelhanças, compartilham suas vocações, seu caráter público, suas similaridades nas estruturas acadêmicas e a equivalência dos níveis de seus serviços. Foi criada em agosto de 1991 para dar resposta aos desafios da sociedade.

A **XIX SEMANA DA ENGENHARIA DE PRODUÇÃO E MECÂNICA SULAMERICANA – XIX SEPROSUL** foi organizada de forma conjunta pelos Programas de Pós Graduação em Engenharia Mecânica (PGMEC) e Engenharia da Produção (PPGEP) da Universidade Federal do Paraná – UFPR. Foi realizada no período de **11 a 13 de novembro de 2019 em Curitiba, PR** - Brasil, abordando como tema principal **“A Nova Indústria e a Contribuição para os Objetivos do Desenvolvimento Sustentável”**.



## XIX SEMANA DA ENGENHARIA DE PRODUÇÃO E MECÂNICA SULAMERICANA: XIX SEPROSUL

Anais do XIX Seprosul – Semana da Engenharia de Produção e Mecânica Sulamericana, 11 a 13 de novembro de 2019, Curitiba / organizado por Carlos Alberto Bavastri, Marcio Brandani Tenório e Natália Gaspar Lamb. - Curitiba, Setor de Tecnologia da UFPR, 2019.

Suporte: E-book  
Formato: PDF

ISBN: 978-85-7335-337-2

1. Tecnologia. 2. Ciências Aplicadas.  
I. Bavastri, C. A., org.; II. Tenório, M. B., org.; III. Lamb, N. G., org. IV.  
Título



# **SEMANA DE ENGENHARIA DE PRODUÇÃO E MECÂNICA SUL-AMERICANA**

10 A 13 DE NOVEMBRO DE 2019 - CURITIBA – PR - BRASIL

## **RESOLUÇÃO DE UM PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO EM UM AMBIENTE FLOWSHOP ATRAVÉS DE UM PROCEDIMENTO METAHEURÍSTICO BIOBJETIVO**

### **SOLVING A PRODUCTION SCHEDULING PROBLEM IN A FLOWSHOP ENVIRONMENT BY MEANS OF A BIOBJECTIVE METAHEURISTIC PROCEDURE**

### **RESOLUCIÓN DE UN PROBLEMA DE PROGRAMACIÓN DE LA PRODUCCIÓN EN UN ENTORNO FLOWSHOP MEDIANTE UN PROCEDIMIENTO METAHEURÍSTICO BIOBJETIVO**

Adrián Andrés Toncovich, Departamento de Ingeniería, Universidad Nacional del Sur,  
atoncovi@uns.edu.ar

Daniel Alejandro Rossit, Departamento de Ingeniería, Universidad Nacional del Sur -  
INMABB UNS CONICET, Departamento de Matemática, daniel.rossit@uns.edu.ar

#### **Resumo**

Neste trabalho analisamos um problema de sequenciamento correspondente a um ambiente de produção do tipo flowshop. Um procedimento metaheurístico biobjetivo é usado para resolver o problema de sequenciamento correspondente à referida configuração produtiva. O problema de programação da produção considerado pode ser caracterizado como um skip flowshop no qual a ligação com os armazéns de matérias-primas e produtos acabados é levada em consideração. Ao avaliar a qualidade das soluções propostas, os objetivos de minimização do tempo total de produção e minimização do atraso total são levados em consideração simultaneamente. O procedimento proposto é baseado em uma estratégia de busca local multiobjetivo que responde à estrutura de metaheurísticas de recozimento simulado. Nesse caso, o método metaheurístico desenvolvido gera um conjunto de soluções que tentam fornecer uma boa aproximação à fronteira ótima de Pareto. A fim de avaliar o desempenho da metodologia proposta, uma série de experimentos são realizados em dois conjuntos de problemas de teste. A partir dos resultados obtidos com o primeiro conjunto de problemas de menor tamanho e sua comparação com um método exato de resolução pode-se afirmar que as soluções fornecidas pelo procedimento proposto são adequadas tanto do ponto de vista de sua qualidade quanto do esforço computacional investido em sua geração. Além disso, o algoritmo foi testado com problemas maiores, para avaliar seu comportamento em espaços de busca mais extensos.

#### **Abstract**

In this work, we analyze a sequencing problem corresponding to a production environment of the flowshop type. A biobjective metaheuristic procedure is used to solve the sequencing problem corresponding to the aforementioned configuration. The production scheduling

problem can be characterized as a skip flowshop in which the links with the warehouses of raw materials and finished products is taken into account. When evaluating the quality of the proposed solutions, the objectives of minimization of maximum completion time (makespan) and minimization of total tardiness are taken into account simultaneously. The proposed procedure is based on a biobjective local search strategy that responds to the structure of simulated annealing metaheuristics. In this case, the metaheuristic method developed generates a set of solutions that try to provide a good approximation to the Pareto optimal front. In order to evaluate the performance of the proposed methodology, a series of experiments are carried out on two sets of test problems. From the results obtained with the first set of problems of smaller size and its comparison with an exact method of resolution, it can be said that the solutions provided by the proposed procedure are suitable both from the point of view of their quality and the computational effort invested in their generation. In addition, the algorithm was tested with larger problems, to evaluate its behavior in wider search spaces.

## Resumen

En este trabajo se analiza un problema de secuenciación correspondiente a un entorno de producción del tipo flowshop. Se recurre a un procedimiento metaheurístico biobjetivo para resolver el problema de secuenciación correspondiente a dicha configuración productiva. El problema de programación de la producción considerado puede caracterizarse como un skip flowshop en el que se tiene en cuenta la vinculación con los almacenes de materias primas y de productos terminados. Al evaluar la calidad de las soluciones propuestas se tienen en cuenta simultáneamente los objetivos de minimización del tiempo de producción total (makespan) y minimización del retraso total (total tardiness). El procedimiento propuesto está basado en una estrategia multiobjetivo de búsqueda local que responde a la estructura de la metaheurística recocido simulado (simulated annealing). En este caso, el método metaheurístico desarrollado genera un conjunto de soluciones que intenta proporcionar una buena aproximación a la frontera óptima de Pareto. Con el fin de evaluar el rendimiento de la metodología propuesta se realizan una serie de experimentos sobre dos conjuntos de problemas de prueba. A partir de los resultados obtenidos con el primer conjunto de problemas de tamaño más reducido y su comparación con un método exacto de resolución puede concluirse que las soluciones proporcionadas por el procedimiento utilizado resultan adecuadas tanto desde el punto de vista de su calidad como del esfuerzo computacional invertido en su generación. Además, el algoritmo fue probado con problemas de mayor tamaño, para evaluar su comportamiento en espacios de búsqueda más amplios.

**Palavras-chave:** SKIP FLOWSHOP; SEQUENCIAMENTO; RECOZIMENTO SIMULADO; OTIMIZAÇÃO MULTI OBJETIVO.

**Keywords:** SKIP FLOWSHOP; SCHEDULING; SIMULATED ANNEALING; MULTIOBJECTIVE OPTIMIZATION.

**Palabras clave:** SKIP FLOWSHOP; PROGRAMACIÓN DE LA PRODUCCIÓN; RECOCIDO SIMULADO; OPTIMIZACIÓN MULTI OBJETIVO.

## **1. Introdução**

El clima de competitividad en el que se desenvuelven las empresas industriales de la actualidad promueve el desarrollo de métodos innovadores para mejorar los procesos de toma de decisiones. Estos métodos están orientados en igual medida ser lo suficientemente rápidos y lograr resultados altamente competitivos. La programación de las operaciones de fabricación es uno de los problemas que frecuentemente se abordan mediante estos métodos. La complejidad asociada con estos problemas limita en gran medida las posibilidades de obtener resultados óptimos, para los casos del mundo real (Błażewicz y otros, 2007; Pinedo, 2012). Considerando los efectos económicos significativos que tiene la programación de las operaciones en el rendimiento de las empresas manufactureras, se impone, ante la dificultad de obtener soluciones óptimas, contar al menos con enfoques adecuados para resolver los problemas con la suficiente celeridad, para obtener soluciones competitivas (Błażewicz y otros, 2007; Pinedo, 2012; Framiñán, Leisten y García, 2014). Los procesos de fabricación comprenden las diferentes operaciones que transforman las materias primas y componentes en productos finales. En general, la formulación general de un problema de programación de la fabricación dado se puede obtener recurriendo a un enfoque de programación matemática (Ruiz, Şerifoğlu y Urlings, 2008; Sayadi, Ramezani y Ghaffari-Nasab, 2010; Defersha y Chen, 2012). En este trabajo se aborda específicamente el problema de programación en un entorno flowshop permutativo (PFSSP, Permutation FlowShop Scheduling Problem). El PFSSP es un problema de optimización combinatoria que pertenece a la clase NP-Completo (Garey, Johnson y Sethi, 1976). El PFSSP se define como el problema de encontrar la mejor programación de trabajos que se deben procesar en un conjunto de máquinas, considerando que todos los trabajos tienen la misma secuencia de procesamiento a través de las máquinas y todas las máquinas procesan los trabajos siguiendo la misma secuencia. Este tipo de configuración suele ser la indicada para obtener mejoras en la productividad industrial y, por ello, se encuentra implementada en un número importante de todas las instalaciones productivas del mundo real (Pan y otros, 2011). Tanto es así que se intenta extender sus beneficios a otros procesos productivos que, por su naturaleza, no se considerarían configuraciones de tipo flowshop. Este es el caso de las celdas de manufactura, que consisten en grupos de máquinas que procesan un conjunto de componentes o artículos que tienen características similares entre sí. Esto permite que dentro de cada celda la configuración sea semejante a un sistema flowshop. Este tipo de estrategias de producción se encuentra ampliamente difundido en la industria (Joines, Culbreth y King, 1996; França y otros, 2005). Sin embargo, las mejoras en la productividad que pueden alcanzarse a partir de su utilización suelen verse comprometidas cuando el entorno industrial en el que operan no está correctamente integrado a la celda. Tal como puede ser el caso del abastecimiento de los materiales para producir cada ítem, o de su almacenamiento una vez finalizada la producción. En este trabajo, se concibe a la celda de manufactura inserta en un sistema productivo, que debe proveer los materiales necesarios para el proceso de manufactura, así como también, recibir y almacenar los ítems terminados.

La medida de eficiencia más frecuentemente utilizada para evaluar la calidad de las soluciones en el PFSSP ha sido el makespan, que se define como el tiempo total de producción requerido para procesar todos los trabajos del sistema. Con el correr del tiempo se han considerado otras medidas de rendimiento y se han añadido diferentes condiciones operativas a la configuración básica. En nuestro caso, el sistema de fabricación está dispuesto en la forma de una célula de fabricación que abarca, además, el suministro de materias primas y la entrega del producto final. El proceso de optimización considera como objetivos el



makespan y la tardanza total en la búsqueda de soluciones para el problema.

En este trabajo se introduce un problema para el cual se propone una formulación de programación matemática y se desarrolla una metodología de solución adecuada que permite abordar su naturaleza combinatoria. La metodología utilizada para resolver el problema se basa en un procedimiento metaheurístico que puede considerarse como una adaptación multiobjetivo del algoritmo de recocido simulado. Este enfoque metaheurístico genera un conjunto de soluciones que permite aproximarse al frente óptimo de Pareto. Se realizan una serie de experimentos para evaluar el rendimiento de la técnica propuesta.

El trabajo está organizado de acuerdo con la siguiente estructura. Primero se presenta el PFSSP y se especifica en detalle su configuración. A continuación, se formula un modelo de programación matemática para el PFSSP. Luego, se presenta el procedimiento de simulated annealing. A continuación, se resuelve un conjunto de problemas utilizando ambos enfoques para comparar los resultados. Posteriormente, el algoritmo de simulated annealing se aplica a instancias más grandes para evaluar su rendimiento. Finalmente, se establecen las conclusiones y se proponen futuras líneas de investigación.

## 2. Caracterización del problema

El problema de programación de la producción introduce una serie de elementos de interés en entornos reales de producción conocidos como células de fabricación (Joines, Culbreth y King, 1996; França y otros, 2005; Hendizadeh y otros, 2008; Li, Li y Gupta, 2015). El problema considerado se limita al análisis de las actividades de programación en un entorno de producción constituido esencialmente por una serie de máquinas que configuran la celda, un almacén de materias primas y un almacén de productos terminados. Entre los almacenes y la celda de manufactura se deben realizar ciertas operaciones de traslado, ya sea de materias primas o de productos terminados, que impactarán en las medidas de desempeño consideradas para resolver el problema de secuenciación. Estas operaciones, naturalmente, dependerán del tipo de producto que se procese (cantidad de piezas, componentes, pesos, volúmenes, etc.).

En primer lugar, se propone una formulación de programación matemática para resolver el problema caracterizado como un sistema flowshop permutativo con la posibilidad de que se tengan trabajos que omitan etapas (Permutation Skip FlowShop), es decir, donde algunos trabajos no necesariamente tienen que ser procesados en todas las máquinas de la celda. Estas características adicionales se encuentran en varios entornos de producción reales. A continuación, se propone el procedimiento metaheurístico multiobjetivo para resolver el problema de programación de la producción (Minella, Ruiz y Ciavotta, 2011; Shoaardebili y Fattahi, 2015).

Las medidas de desempeño utilizadas para evaluar la calidad de las secuencias tienen en cuenta el aprovechamiento de los recursos de producción y las desviaciones con respecto a las fechas de entrega. En particular, se tienen en cuenta dos objetivos al analizar la calidad de las soluciones candidatas: la minimización del tiempo requerido para el procesamiento de todos los trabajos (makespan) y la minimización de la tardanza total (total tardiness). La Figura 1 muestra un ejemplo de configuración de una celda de fabricación flowshop constituida por tres máquinas ( $M_1$ ,  $M_2$  y  $M_3$ ), un almacén de materias primas ( $A_{MP}$ ) y un almacén de productos terminados ( $A_{PF}$ ).

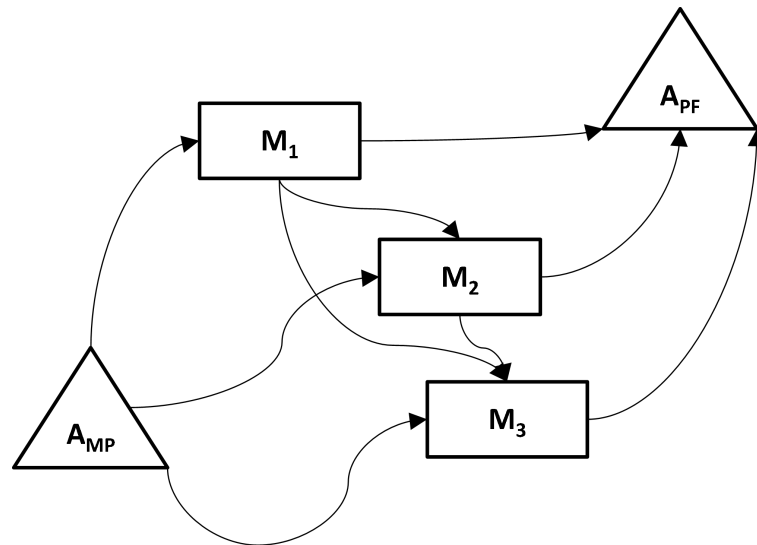


FIGURA 1 - Ejemplo de configuración de una celda de fabricación con tres máquinas.

### 3. Modelo de programación matemática

El problema abordado en este trabajo es un problema de secuenciación en una celda de fabricación que se encuentra alejada de los almacenes de materias primas y de productos terminados. Los trabajos tienen una fecha de liberación, y una vez producida esta, las materias primas requeridas para procesar el trabajo pueden ser transportadas desde el almacén de materias primas hasta la celda de fabricación. Cuando las materias primas necesarias para realizar el trabajo llegan a la célula de fabricación, puede comenzar su procesamiento. La configuración productiva dentro de la celda de fabricación es del tipo flowshop, por lo tanto, todos los trabajos tienen la misma secuencia de máquinas. Sin embargo, algunos de los trabajos no requieren ser procesados en todas las máquinas, es decir, pueden omitir algunas operaciones. Después de que cada trabajo ha terminado de procesarse, debe ser transportado desde la célula de fabricación hasta el almacén de productos terminados. Este último tiempo de transporte repercute en el tiempo de entrega del trabajo, ya que debe añadirse al tiempo de finalización del trabajo en la celda de fabricación.

Otros supuestos que se hacen para esta formulación del problema son:

- Cada trabajo puede ser procesado por una sola máquina a la vez.
- Una vez que se inicia el procesamiento de un trabajo, no puede interrumpirse.
- Las máquinas de la celda pueden estar ociosas durante el horizonte de planificación.

#### 3.1 Conjuntos

$j$ : índice para trabajos,  $j = 1, 2, \dots, n$ .

$m$ : índice para máquinas,  $m = 1, 2, \dots, m$ .

#### 3.2 Parámetros

$p_{j,m}$ : tiempo de procesamiento del trabajo  $j$  en la máquina  $m$ .

$t_{MP_j}$ : tiempo de transporte de materias primas correspondientes al trabajo  $j$  hasta la célula de fabricación.

$r_j$ : liberación de la orden del trabajo  $j$ .

$d_j$ : fecha de entrega del trabajo  $j$ .

$t_{PF_j}$ : tiempo de transporte de producto terminado correspondiente al trabajo  $j$  al almacén.

$M$ : un número positivo suficientemente grande.

### 3.3 Variables

$C_{j,m}$ : tiempo de finalización del trabajo  $j$  en la máquina  $m$ .

$T_j$ : tardanza del trabajo  $j$ .

$C_{m\acute{a}x}$ : tiempo total de producción (makespan).

$x_{j',j}$ : variable binaria, vale 1 si el trabajo  $j'$  es procesado antes que el trabajo  $j$ , y 0 en otro caso.

### 3.4 Función objetivo

Se busca minimizar el tiempo total de producción (makespan) y la tardanza total (tardiness).

$$\text{Min } z = C_{m\acute{a}x} + \sum_j T_j \quad (1)$$

### 3.5 Restricciones

Restricción de precedencia (2), asegura que el trabajo  $j$  no comience su procesamiento en la máquina  $m$  antes de que haya terminado su procesamiento en la máquina  $m-1$ .

$$C_{j,m} \geq C_{j,m-1} + p_{j,m}; \forall j, m > 1 \quad (2)$$

Restricción de ordenamiento (3), establece que el trabajo  $j$  no comience su procesamiento en la máquina  $m$  si antes no terminaron su procesamiento todos los trabajos  $j'$  precedentes.

$$C_{j,m} \geq C_{j',m} + p_{j,m} - (1 - x_{j',j}) \cdot M; \forall m, j' \neq j \quad (3)$$

Restricción lógica (4), si el trabajo  $j'$  está secuenciado antes que el trabajo  $j$ , entonces lo inverso no es válido.

$$x_{j',j} + x_{j,j'} = 1; j \neq j' \quad (4)$$

Abastecimiento de materias primas (5), cada trabajo no puede comenzar su procesamiento, si antes no fue liberada la orden correspondiente y las materias primas no fueron suministradas.

$$C_{j,m} \geq r_j + t_{MP_j} + p_{j,m}; m = 1, \forall j \quad (5)$$

Makespan (6), para calcular la finalización del trabajo  $j$ , resulta necesario considerar el tiempo de traslado hacia el almacén de productos finalizados.

$$C_{m\acute{a}x} \geq C_{j,m} + t_{PF_j} ; \forall j, \forall m \quad (6)$$

Tardanza (7), para calcular la tardanza del trabajo  $j$ , es necesario considerar el tiempo de traslado hasta el almacén de producto terminado.

$$T_j = \text{máx} \{0, (C_{j,m} + t_{PF_j}) - d_j\} ; \forall j, m = M \quad (7)$$

Restricciones (8) relacionadas con los valores no negativos y binarios de las variables.

$$T_j, C_{j,m} \geq 0; x_{jj'} \in \{0,1\}; \forall j, \forall m \quad (8)$$

## 4. Solución metaheurística basada en simulated annealing

En esta sección se describe el algoritmo basado en simulated annealing (SA) utilizado para resolver el problema presentado anteriormente. El algoritmo tiene la estructura de un procedimiento de simulated annealing con archivo de soluciones de Pareto (PASA, Pareto Archived Simulated Annealing). SA es un método de búsqueda local que se desarrolló a partir de una analogía con el fenómeno de recocido (Kirkpatrick, Gelatt y Vecchi, 1983) para resolver problemas complejos de optimización. Los métodos de búsqueda local buscan la solución con el mejor valor del objetivo elegido en el entorno de la solución actual, la aceptan como la solución actual y repiten este procedimiento hasta que no resulte posible mejorar la solución en el entorno explorado. Mediante la aplicación sistemática de este procedimiento se obtiene, en general, un óptimo local para el problema. Para evitar quedar atrapado en un óptimo local, se debe incorporar un mecanismo de diversificación con el fin de explorar adecuadamente el espacio de soluciones. En la metaheurística simulated annealing, la estrategia de diversificación permite movimientos, con cierta probabilidad, hacia soluciones que empeoran el valor actual de la función objetivo. El recocido simulado ya ha demostrado su capacidad de resolución de problemas en entornos productivos flowshop regulares (Osman y Potts, 1989; Low, 2005; Vahedi Nouri, Fattahi y Ramezani, 2013).

### 4.1 Procedimiento PASA

Serafini (1992) y Fortemps, Teghem y Ulungu (1994) propusieron dos procedimientos similares basados en el recocido simulado para resolver problemas de optimización combinatoria multiobjetivo. Posteriormente, Czyzak y Jaszkievicz (1998) y Engrand y Mouney (1998) desarrollaron otras variantes de la metodología que consideran de forma más explícita las características multiobjetivo del problema (Collette y Siarry, 2004). En este trabajo, siguiendo lo propuesto por Engrand y Mouney (1998), se utiliza una función de agregación de las funciones objetivo, junto con un sistema de archivo de soluciones no dominadas encontradas durante el proceso de búsqueda. Se supone que las funciones objetivo a minimizar,  $f_p$ ,  $p = 1, 2, \dots, P$ , son positivas. En tal caso, el problema se puede transformar en un problema de minimización de un único objetivo utilizando la siguiente función de agregación (9).

$$z(S) = \sum_{p=1}^P \ln(f_p(S)) \quad (9)$$

Donde  $S$  es una secuencia dada. En consecuencia, la expresión (10) representa la

variación relativa media de las funciones objetivo entre la solución actual ( $S_A$ ) y la solución candidata ( $S_C$ ).

$$\Delta z = z(S_C) - z(S_A) = \sum_{p=1}^P \ln \left( \frac{f_p(S_C)}{f_p(S_A)} \right) \quad (10)$$

Si  $\Delta z > 0$ ,  $S_C$  deteriora el valor medio relativo del conjunto de funciones objetivo. Si  $\Delta z < 0$ ,  $S_C$  mejora el valor medio relativo del conjunto de funciones objetivo. En el primer caso se acepta la solución  $S_C$  con una probabilidad dada por  $PA = e^{-\Delta z/T}$ , donde  $T$  es el parámetro de control que simula la temperatura en el proceso metalúrgico del recocido. El método tiene en cuenta un archivo de soluciones no dominadas que se gestiona de la siguiente manera:

- Si  $S_C$  está dominada por al menos alguna de las soluciones del archivo, no se incorpora al conjunto.
- Si  $S_C$  domina a una o más soluciones en el archivo, es incorporada eliminando las soluciones que domina.
- Si  $S_C$  no domina, ni es dominada por las soluciones del archivo, es incorporada sin eliminar ninguna solución.

Para obtener una aproximación a toda la frontera eficiente durante el proceso de búsqueda, es necesario reiniciar la búsqueda regularmente desde una de las soluciones archivadas seleccionada al azar. El algoritmo PASA incorpora los parámetros clásicos del recocido simulado:

$T$ : parámetro de control (temperatura), valor positivo que varía desde un valor inicial mayor,  $T_0$ , a otro que es menor,  $T_f$ , durante la ejecución del algoritmo.

$N_T$ : número de iteraciones realizadas por el algoritmo para cierto valor de  $T$ .

$\alpha$ : función en  $T$ ,  $\alpha = \alpha(T)$ , que determina la variación de  $T$ . En general  $\alpha(T) = \alpha \cdot T$ , en la práctica  $\alpha \in [0,8; 0,9999]$ .

$N_{stop}$ : número máximo de iteraciones permitidas sin mejora.

Se aplica el siguiente pseudocódigo de la Figura 2 para determinar un conjunto de soluciones potencialmente eficientes (óptimas de Pareto).

Resulta necesario consignar que la naturaleza multiobjetivo del algoritmo no fue específicamente probada en este trabajo. Es decir, para comparar los resultados de la metaheurística con los de la programación matemática solo se consideraron las mejores soluciones en términos de la función objetivo dada por la ecuación (1).

## 4. Experimentación computacional

Para evaluar los resultados de la formulación de programación matemática y del algoritmo propuesto en este trabajo, se consideraron instancias de diferentes tamaños. El número de trabajos se definió a partir de los valores 5, 10, 15 y 20, mientras que para el número de máquinas se tuvieron en cuenta las mismas posibilidades (5, 10, 15 y 20). No todas las combinaciones posibles fueron consideradas. La Tabla 1 presenta las instancias evaluadas.

## i. Inicio

Se utiliza un procedimiento constructivo aleatorizado para generar una solución inicial,  $S_0$ .

Se evalúa  $f_p(S_0)$ ,  $\forall p$ .

$S_0$  se incorpora al conjunto de soluciones eficientes:  $CE = \{S_0\}$ ,  $N_{cont} = t = 0$ ,  $T = T_0$ .

## ii. Iteración t

Se genera aleatoriamente una solución en el entorno de  $S_A$ ,  $S_C \in V(S_A)$ .

Se evalúa  $f_p(S_C)$ ,  $\forall p$ .

Se calcula  $\Delta_z = z(S_C) - z(S_A) = \sum_p \ln(f_p(S_C)/f_p(S_A))$ .

Si  $\Delta_z \leq 0$ , la nueva solución es aceptada:  $S_A \leftarrow S_C$ ,  $N_{cont} = 0$ .

En otro caso,  $S_C$  es aceptada con la siguiente probabilidad  $PA = e^{-\Delta_z/T}$ .

Se genera un número aleatorio  $\xi$  uniformemente distribuido en el intervalo  $[0; 1]$ :

- si  $\xi \leq PA$ ,  $S_A \leftarrow S_C$ ,  $N_{cont} = 0$ .
- si  $\xi > PA$ ,  $S_A \leftarrow S_A$ ,  $N_{cont} = N_{cont} + 1$ .

Si corresponde, se actualiza  $CE$  teniendo en cuenta  $S_C$ .

$t \leftarrow t + 1$ : Si  $t$  es un múltiplo de  $N_T$ , entonces  $T = \alpha \cdot T$ , en otro caso el valor de  $T$  se mantiene. Si  $N_{cont} = N_{stop}$  o  $T < T_f$ , se detiene la ejecución, en otro caso se continúa la ejecución.

FIGURA 2 – Pseudocódigo del procedimiento PASA.

De acuerdo con lo indicado al introducir la formulación de programación matemática, resulta necesario definir los valores de varios parámetros para especificar los cinco escenarios considerados dentro de cada instancia. Para el caso de los tiempos de procesamiento,  $p_{j,m}$ , los datos se obtuvieron a partir de una distribución cuasi-uniforme  $[0; 100]$ . Dado que el problema abordado contempla la posibilidad de que existan operaciones omitidas, se incluye el valor 0 en la distribución del tiempo de procesamiento, y como característica particular, la probabilidad de omitir operaciones se considera del 3 %, es decir, no todos los valores posibles tienen la misma probabilidad, es por ello se habla de una distribución cuasi-uniforme. Esto amplifica el impacto de las operaciones omitidas en el problema de programación. Para los tiempos de transporte desde el almacén de materias primas hasta la celda de fabricación y desde la celda de fabricación hasta el almacén de productos finalizados, los datos se tomaron de una distribución uniforme  $[10; 20]$ . Las fechas de liberación de los trabajos se obtuvieron a partir de una distribución uniforme  $[1; 100]$ . Respecto a la fecha de vencimiento de los trabajos, se siguieron los lineamientos dados en Ruiz y Stützle (2008), sin embargo en ese artículo la fecha de vencimiento se calcula de la siguiente manera:  $d_j = r_j + \sum_m p_{j,m} \cdot (1 + aleatorio \cdot 3)$ . Esta representación incluye todos los tiempos de procesamiento que el trabajo  $j$  debe recorrer para su finalización, una vez producido su lanzamiento, más un tiempo adicional dado por el segundo término de la expresión encerrada entre paréntesis, que



considera un número al azar (*aleatorio*) aleatorio entre  $[0; 1]$ . En este caso, deben tenerse en cuenta para cada trabajo otros tiempos adicionales a los ya mencionados, es decir, los tiempos de transporte desde el almacén de materias primas hasta la célula de fabricación y desde la célula hasta el almacén de producto terminado. Por lo tanto, en este caso la fecha de entrega del trabajo  $j$ , se calculará como:  $d_j = r_j + t_{MP_j} + t_{PF_j} + \sum_m p_{j,m} \cdot (1 + \text{aleatorio} \cdot 0,3)$ . Además, en la presente formulación, en el cálculo de la fecha de entrega la constante que multiplica el número aleatorio, se ha reducido de 3 a 0,3. Esto genera las fechas de entrega más ajustadas, dando una mayor relevancia en el enfoque multiobjetivo a la tardanza total con respecto al makespan. Para cada instancia (combinación de número de máquinas y número de trabajos) se generaron cinco conjuntos de parámetros diferentes.

Tabla 1 - Configuración de las instancias evaluadas.

Instancia	Número de trabajos	Número de máquinas
1	5	5
2	5	10
3	10	5
4	10	10
5	15	10
6	15	15
7	20	10

Para los casos resueltos mediante el enfoque de programación matemática se empleó Pyomo (Hart, Watson y Woodruff, 2011; Hart y otros, 2012), como lenguaje de modelado, y CPLEX 12.5.0 como software de optimización. Las pruebas se ejecutaron en una CPU con 8 GB de RAM e i-core 5, el sistema operativo es de 64 bits y el tiempo de ejecución límite fue establecido en una hora (3600 segundos). CPLEX pudo resolver instancias de hasta 10 trabajos y 10 máquinas. Para la siguiente instancia, 15 trabajos y 10 máquinas, no resultó posible encontrar una solución con una desviación (GAP) inferior al 3 %. En promedio la brecha reportada por CPLEX después de una hora de corrida fue mayor al 30 %. Estas soluciones fueron descartadas. Los casos resueltos utilizando el algoritmo PASA fueron programados en VBA (Visual Basic for Applications) como lenguaje de modelado sobre una configuración de MS-Excel. Las pruebas se ejecutaron en una CPU con 8 GB de RAM e i-core 7 y un sistema operativo de 64 bits.

## 5. Resultados

La Tabla 2 presenta los resultados de CPLEX y del algoritmo PASA. Los valores informados para el algoritmo PASA corresponden al promedio de 30 corridas y el valor de la mejor corrida para cada escenario. La tabla muestra el valor de la función objetivo global (FO), es decir, la suma de ambos objetivos y, a continuación, el valor de makespan ( $C_{\max}$ ), la tardanza total (TT), y el tiempo, medido en segundos, requerido para obtener la mejor solución, respectivamente. Para el primer grupo de escenarios (5T\_5M) se puede ver que el algoritmo PASA funciona bastante bien, obteniendo para todos los casos la solución óptima (el valor medio coincide con el mejor valor). Esta precisión del algoritmo PASA se repite para el segundo grupo de escenarios (5T\_10M). Para los siguientes grupos de escenarios, tenemos ligeras diferencias entre el valor de las funciones objetivo. Para tener una mejor percepción de

esto, se presenta la Tabla 3.

En la Tabla 3, la comparación entre la solución obtenida mediante CPLEX y la solución generada a través de PASA se expresa en términos de la diferencia porcentual entre los valores correspondientes, con respecto a la solución obtenida al resolver con CPLEX. A partir de la Tabla 3, es posible ver que todos los porcentajes están por debajo del 1 %, además, muchos de ellos son directamente 0 %, lo cual quiere decir que la solución óptima y la solución media encontrada por el algoritmo PASA coinciden. A pesar de que la magnitud del error es relativamente menor, cabe mencionar que el algoritmo PASA está equilibrado en la búsqueda de ambas funciones objetivo. Concretamente, para los casos de 10 trabajos y 5 máquinas el error porcentual para el objetivo del tiempo total de fabricación es menor que el error para el objetivo de tardanza. Mientras tanto, para los casos de 10 trabajos y 10 máquinas, ocurre lo contrario. Por lo tanto, el algoritmo PASA no está sesgado en la búsqueda de uno de los objetivos con respecto al otro, tal como se pretende.

Tabla 2 - Resultados de CPLEX y del algoritmo PASA.

Instancia	Escenario	MIP-CPLEX				PASA							
						Promedio (30 corridas)				Mejor valor obtenido			
		FO	C <sub>máx</sub>	TT	Tiempo	FO	C <sub>máx</sub>	TT	Tiempo	FO	C <sub>máx</sub>	TT	Tiempo
5T_5M	1	970	510	460	< 1	970	510	460	< 1	970	510	460	< 1
	2	834	553	281	< 1	834	553	281	1	834	553	281	< 1
	3	951	517	434	< 1	951	517	434	1	951	517	434	< 1
	4	1201	621	580	< 1	1201	621	580	1	1201	621	580	< 1
	5	1156	591	565	< 1	1156	591	565	< 1	1156	591	565	< 1
5T_10M	1	1394	929	465	1	1394	929	465	1	1394	929	465	< 1
	2	1431	835	596	1	1431	835	596	1	1431	835	596	< 1
	3	1082	847	235	1	1082	847	235	< 1	1082	847	235	< 1
	4	1210	746	464	1	1210	746	464	1	1210	746	464	< 1
	5	1228	895	333	1	1228	895	333	1	1228	895	333	< 1
10T_5M	1	2742	845	1897	10	2757	843	1914	23	2742	845	1897	8
	2	2579	791	1788	10	2586	792	1795	21	2579	791	1788	< 1
	3	2882	902	1980	9	2898	914	1983	18	2882	902	1980	< 1
	4	2361	775	1586	12	2375	777	1598	30	2361	775	1586	8
	5	2903	755	2148	10	2909	756	2153	29	2903	755	2148	5
10T_10M	1	2746	1095	1651	22	2773	1097	1676	27	2746	1095	1651	< 1
	2	3358	1158	2200	17	3365	1160	2205	25	3358	1158	2200	7
	3	2560	1058	1502	19	2566	1060	1506	21	2560	1058	1502	< 1
	4	2571	1040	1531	23	2611	1077	1534	34	2571	1040	1531	10
	5	3262	1060	2202	14	3265	1063	2202	18	3262	1060	2202	< 1

Los casos más grandes fueron resueltos solo por el algoritmo PASA, ya que CPLEX no fue capaz de encontrar una solución reportable en un tiempo de cómputo de una hora. En la Tabla 4, se puede ver la precisión del algoritmo PASA, ya que la diferencia entre el mejor



valor con respecto al valor promedio (considerando 30 corridas para cada escenario) es relativamente pequeña. Para un análisis directo de esta cuestión se presenta la Tabla 5. En la Tabla 5 se muestra la diferencia porcentual entre el mejor valor obtenido por el algoritmo PASA y el promedio obtenido por PASA después de 30 corridas. Estos valores indican que el algoritmo se comporta de manera estable y replicable, capaz de producir resultados similares en condiciones similares. Este aspecto constituye una fortaleza del algoritmo. La diferencia media entre el mejor valor y el valor medio de la función objetivo global es inferior al 4 %, apoyando la capacidad del algoritmo para producir soluciones de alta calidad empleando pocas corridas. Para el objetivo del makespan la diferencia entre el mejor valor y el valor medio es del 2,4 % y para el objetivo de la tardanza es del 4,4 %.

Tabla 3 - Comparación entre las soluciones promedio de CPLEX y del algoritmo PASA.

Instancia	CPLEX			PASA					
	FO	C <sub>máx</sub>	TT	FO		C <sub>máx</sub>		TT	
				Promedio	Diferencia	Promedio	Diferencia	Promedio	Diferencia
5T_5M	1022	558	464	1022	0,00 %	558	0,0 %	464	0,0 %
5T_10M	1269	850	418	1269	0,00 %	850	0,0 %	419	0,0 %
10T_5M	2693	813	1879	2705	-0,45 %	816	-0,6 %	1889	-1,1 %
10T_10M	2899	1082	1817	2916	-0,59 %	1091	-0,8 %	1825	-0,6 %

Tabla 4 - Instancias más grandes resueltas por el algoritmo PASA.

Instancia	Escenario	PASA							
		Promedio (30 corridas)				Mejor valor obtenido			
		FO	C <sub>máx</sub>	TT	Tiempo	FO	C <sub>máx</sub>	TT	Tiempo
15T_10M	1	6633	1503	5130	67	6370	1441	4929	71
	2	6520	1504	5016	57	6306	1499	4807	71
	3	6375	1444	4931	69	6173	1423	4750	55
	4	6109	1470	4639	58	5923	1400	4523	113
	5	6044	1480	4564	52	5789	1437	4352	114
15T_15M	1	5794	1714	4080	84	5562	1658	3904	105
	2	7452	1839	5613	99	7347	1705	5642	275
	3	7230	1859	5371	133	6983	1866	5117	140
	4	7497	1805	5693	145	7152	1804	5348	180
	5	6622	1753	4869	166	6416	1774	4642	191
20T_10M	1	11349	1789	9560	202	10957	1779	9178	177
	2	11286	1799	9486	172	10798	1789	9009	107
	3	10463	1636	8827	191	9846	1602	8244	157
	4	10830	1751	9078	143	10412	1711	8701	156
	5	11248	1801	9448	106	10649	1696	8953	101

## 6. Conclusión

En este trabajo se estudió un caso particular del PFSSP (Permutation FlowShop

Scheduling Problem), que incluye características tales como: dos medidas de desempeño, tiempos de transporte, omisión de operaciones, fechas de lanzamiento y fechas de entrega variables. Para este problema se presentaron una formulación de programación matemática y un algoritmo de recocido simulado multi-objetivo (Algoritmo PASA). Los resultados de ambos enfoques son similares en cuanto a su calidad. Sus tiempos de ejecución, probados en problemas de ejemplo, difieren significativamente tan pronto como el tamaño del problema aumenta, con un mejor desempeño del enfoque meta-heurístico. Problemas de mayor tamaño también fueron abordados mediante el algoritmo PASA. Se observó que los resultados obtenidos eran estables en términos de precisión y las soluciones se obtuvieron en un tiempo de cálculo razonable.

Tabla 5 - Precisión del algoritmo PASA.

Instancia	Escenario	FO	C <sub>máx</sub>	TT
15T_10M	1	4,1 %	4,3 %	4,1 %
	2	3,4 %	0,3 %	4,4 %
	3	3,3 %	1,5 %	3,8 %
	4	3,2 %	5,0 %	2,6 %
	5	4,4 %	3,0 %	4,9 %
15T_15M	1	4,2 %	3,4 %	4,5 %
	2	1,4 %	7,8 %	-0,5 %
	3	3,5 %	-0,4 %	5,0 %
	4	4,8 %	0,0 %	6,4 %
	5	3,2 %	-1,2 %	4,9 %
20T_10M	1	3,6 %	0,6 %	4,2 %
	2	4,5 %	0,6 %	5,3 %
	3	6,3 %	2,1 %	7,0 %
	4	4,0 %	2,4 %	4,3 %
	5	5,6 %	6,2 %	5,5 %
	Promedio	4,0 %	2,4 %	4,4 %

En futuras investigaciones se considerarán instancias de mayor tamaño y otras medidas de desempeño, realizando comparaciones de resultados con estrategias de solución alternativas, tratando de realizar contribuciones en el desarrollo de método precisos y eficientes para resolver este tipo de problemas de programación de la producción.

## Referencias

- BŁAŻEWICZ, J.; ECKER, K.H.; PESCH, E.; SCHMIDT, G.; WEGLARZ, J. *Handbook on scheduling: from theory to applications*. Berlin Heidelberg: Springer, 2007.
- PINEDO, M. *Scheduling: theory, algorithms, and systems*. New Jersey: Prentice-Hall, 2012.
- FRAMIÑÁN, J.M.; LEISTEN, R.; GARCÍA, R.R. *Manufacturing scheduling systems*. London: Springer, 2014.
- RUIZ, R.; ŞERİFOĞLU, F.S.; URLINGS, T. Modeling realistic hybrid flexible flowshop scheduling problems. *Computers and Operations Research*, v.35, n.4, p.1151-1175, 2008.
- SAYADI, M.; RAMEZANIAN, R.; GHAFARI-NASAB, N. A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, v.1, n.1, p. 1-10, 2010.

- DEFERSHA, F.M.; CHEN, M. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem. *International Journal of Advanced Manufacturing Technology*, v.62, n.1, p.249-265, 2012.
- GAREY, M.R.; JOHNSON, D.S.; SETHI, R. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, v.1, n.2, p.117-129, 1976.
- PAN, Q.K.; TASGETIREN, M.F.; SUGANTHAN, P.N.; CHUA, T.J. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. *Information Sciences*, v.181, n.12, p. 2455-2468, 2011.
- JOINES, J.A.; CULBRETH, C.T.; KING, R.E. Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transactions*, v.28, n.1, p.69-85, 1996.
- FRANÇA, P.M.; GUPTA, J.N.; MENDES, A.S.; MOSCATO, P.; VELTINK, K.J. Evolutionary algorithms for scheduling a flowshop manufacturing cell with sequence dependent family setups. *Computers and Industrial Engineering*, v.48, n.3, p.491-506, 2005.
- HENDIZADEH, S.H.; FARAMARZI, H.; MANSOURI, S.A.; GUPTA, J.N.; ELMEKKAWY, T.Y. Meta-heuristics for scheduling a flowline manufacturing cell with sequence dependent family setup times. *International Journal of Production Economics*, v.111, n.2, p.593-605, 2008.
- LI, Y.; LI, X.; GUPTA, J.N. Solving the multi-objective flowline manufacturing cell scheduling problem by hybrid harmony search. *Expert Systems with Applications*, v.42, n.3, p. 1409-1417, 2015.
- MINELLA, G.; RUIZ, R.; CIAVOTTA, M. Restarted iterated Pareto greedy algorithm for multi-objective flowshop scheduling problems. *Computers and Operations Research*, v.38, n.11, p.1521-1533, 2011.
- SHOARDEBILI, N.; FATTAHI, P. Multi-objective meta-heuristics to solve three-stage assembly flow shop scheduling problem with machine availability constraints. *International Journal of Production Research*, v.53, n.3, p.944-968, 2015.
- KIRKPATRICK, S.; GELATT, C.D.; VECCHI, M.P. Optimization by simulated annealing. *Science*, v.220, n.4598, p.671-680, 1983.
- OSMAN, I.H.; POTTS, C.N. Simulated annealing for permutation flow-shop scheduling. *Omega*, v.17, n.6, p.551-557, 1989.
- LOW, C. Simulated annealing heuristic for flow-shop scheduling problems with unrelated parallel machines. *Computers and Operations Research*, v.32, n.8, p.2013-2025, 2005.
- VAHEDI NOURI, B.; FATTAHI, P.; RAMEZANIAN, R. Hybrid firefly-simulated annealing algorithm for the flow-shop problem with learning effects and flexible maintenance activities. *International Journal of Production Research*, v.51, n.12, p.3501-3515, 2013.
- SERAFINI, P. Simulated Annealing for Multi Objective Optimization Problems. In: TZENG, G.H.; WANG, H.F.; WEN, U.P.; YU, P.L. (org.). *Multiple criteria decision making: expand and enrich the domains of thinking and application*. p.283-292. New York: Springer, 1992.
- FORTEMPS, P.; TEGHEM, J.; ULUNGU, B. Heuristics for multiobjective combinatorial optimization by simulated annealing. In: *Proceedings of the XIth International Conference on Multiple Criteria Decision Making*. University of Coimbra, Portugal, 1994.
- CZYZAK, P.; JASZKIEWICZ, A. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, v.7, n.1, p.34-47, 1998.
- ENGRAND, P.; MOUNEY, X. Une méthode originale d'optimisation multiobjectif. Technical Report HT-14/97/035/A (98NJ00005). *EDF-DER Electricité de France, Direction des Études et Recherches*. Clamart, France, 1998.
- COLLETTE, Y.; SIARRY, P. Multiobjective methods using metaheuristics. In: COLLETTE, Y.; SIARRY, P. (org.). *Multiobjective Optimization: Principles and Case Studies*. p.109-134. Berlin Heidelberg: Springer, 2004.
- RUIZ, R.; STÜTZLE, T. An iterated greedy heuristic for the sequence dependent setup times flow-shop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, v.187, n.3, p.1143-1159, 2008.



# SEMANA DE ENGENHARIA DE PRODUÇÃO E MECÂNICA SUL-AMERICANA

10 A 13 DE NOVEMBRO DE 2019 - CURITIBA – PR - BRASIL

HART, W.E.; WATSON, J.P.; WOODRUFF, D.L. Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, v.3, n.3, p. 219-260, 2011.

HART, W.E.; LAIRD, C.; WATSON, J.P.; WOODRUFF, D.L. *Pyomo—optimization modeling in python*. New York :Springer, 2012.

