

Trabajo de grado en modalidad de aplicación

[173030] Diseño de una metaheurística GRASP hibridizada con la metodología PAES y la simulación de Monte Carlo en un ambiente *Flexible Flow Shop* estocástico multi-objetivo.

Silvia Juliana Castro Mesa^{a,c}, Paola di Gregorio Martínez^{a,c}, Laura Isabel Ortiz Sarria^{a,c},

Ing. Eliana María González Neira, M.Sc^{b,c}

^aEstudiante de Ingeniería Industrial

^bProfesor, Director del Proyecto de Grado, Departamento de Ingeniería Industrial

^cPontificia Universidad Javeriana, Bogotá, Colombia

Resumen de diseño en Ingeniería

To achieve a higher level of efficiency within a manufacturing industry, the production scheduling is essential, because this process is crucial for the maximization of the business value. Currently, a big part of literature in scheduling is focused on solving a deterministic problem to minimize the makespan. Given that, realistically, the industry is exposed to random events that can affect its performance, the aim of this project is to study a multi-objective stochastic *Flexible Flow Shop (FFS)* environment. The objectives to minimize are expected value of tardiness, standard deviation of tardiness, expected value of total completion time (equal to flowtime due to release times are zero) and standard deviation of total completion time. The stochastic parameters are the times between failures and times to repair the machines (duration of machine breakdowns). As solution method, a simheuristic is proposed, which hybridizes the metaheuristic *Greedy Randomized Adaptive Search Procedures (GRASP)* with the Monte Carlo simulation and Pareto Archived Evolution Strategy (PAES) algorithm to obtain the Pareto frontier (see illustration 2). A first experimental design is done to test the simheuristic performance for the deterministic version (see illustration 1) of the problem by comparing the results of the simheuristic with the flowtime obtained by scheduling the jobs with FL dispatching rule, and the tardiness with the ENS2 dispatching rule. A second design of experiments is designed to evaluate the effects of different coefficients of variation and probability distribution of both stochastic parameters in the four objective functions of the stochastic case. To do both experimental designs 324 benchmark instances were evaluated in both cases. Results show, that for the deterministic case, the metaheuristic presents an average improvement of 3% in flowtime against FL rule, 2% in tardiness against ENS2 rule. For the stochastic case, results show that both probability distributions and coefficient of variation have a significant effect in the four response variables, which shows the importance of an accurate fitting of probability distributions to obtain adequate solutions.

Key Words: Scheduling, Flexible Flow Shop (FFS), GRASP, PAES, Mean Time Between Failures (MTBF), Medium Time to Repair (MTTR), ENS2, FL, flowtime, tardiness, Monte Carlo simulation.

1. Justificación y planteamiento del problema

En la actualidad, por causa de la creciente competitividad del mercado global del siglo XX, las empresas han buscado la forma de generar un valor agregado para el cliente, razón por la cual, los departamentos de producción se han enfocado en generar un aporte desde su área a este objetivo organizacional, mejorando así, sus operaciones y procesos. Con base en esto, la programación de la producción (en inglés *scheduling*), es una herramienta analítica de apoyo a la toma de decisiones, que ha adquirido un papel fundamental en las industrias manufactureras y de servicios, siendo para esta primera industria una cuestión de mucha importancia, debatida por economistas para una base sólida del desarrollo industrial (Nahmias, 2007). La programación de la producción consiste en asignar recursos a ciertas tareas en determinados periodos de tiempo, con el propósito

de optimizar uno o más objetivos (Pinedo, 2016). Estos pueden ser la reducción de los tiempos totales de procesamiento, los cambios y tiempos de alistamiento y tardanzas, la continuidad y productividad de las operaciones o la reducción de los desperdicios, entre otros.

Los problemas de *scheduling* se pueden denotar por una tripleta alfa, beta y gamma (α, β, γ), propuesta por Graham, Lawler, Lenstra, & Rinnooy Kan (1979). El campo alfa (α) describe el ambiente de máquina y contiene solamente una entrada. El campo beta (β) proporciona detalles de las características de procesamiento y restricciones y puede no contener entradas en absoluto, una sola entrada o múltiples entradas. El campo gamma (γ) describe el objetivo a ser minimizado, y a menudo contiene una sola entrada (Pinedo, 2016). De acuerdo con Pinedo (2016), los posibles entornos de la máquina en el campo alfa (α) son: *Single machine*, *Parallel Machines*, *Flow shop (FS)*, *Flexible flow shop (FFS)*, *Permutation flow shop (PFS)*, *Job shop*, *Flexible job shop*, *Open shop*.

Las fábricas en las que las operaciones están dispuestas en serie, los trabajos deben seguir una misma ruta, primero a la operación 1, luego a la 2 y así sucesivamente, hasta la última operación, corresponden a un ambiente FS o FFS. El estudio de los ambientes FS y FFS ha tenido tanto auge porque su aplicación está dada en un diverso número de industrias como las líneas de ensamblaje automotriz, química, petroquímica, acero inoxidable, alimentos, entre otros (Rabiee, Jolai, Asefi, Fattahi, & Lim, 2016; Tang-Liu & Acheampong, 2005).

El problema FS estándar se define como el conjunto N de n trabajos que deben ser procesados en un conjunto M de m máquinas. Estas m máquinas están dispuestas en serie y cada trabajo $j \in N_j$, se divide en m tareas, una por máquina. Un trabajo modela un determinado lote de producción de un producto o una orden del cliente que debe fabricarse (Q.-K. Pan & Ruiz, 2012). La diferencia del FS con el FFS es que, en el problema FFS existen para cada tarea (u operación) un conjunto de máquinas en paralelo que pueden hacerla, llamado estación. Así el FFS se define como un conjunto de S estaciones dispuestas en serie, en donde al menos una de las estaciones contiene dos o más máquinas en paralelo. Cada trabajo j debe ser procesado en una y sólo una de las máquinas de cada estación (Ruiz & Vázquez-Rodríguez, 2010).

La mayoría de los problemas en FS y FFS en la literatura, han sido estudiados desde el punto de vista determinístico, es decir, todos los parámetros son conocidos y no tienen variaciones (Elyasi & Salmasi, 2013b). Como por ejemplo, el trabajo propuesto por Vallada, Ruiz, & Minella (2008) en donde se buscó minimizar la tardanza total para un ambiente FS determinístico, realizando una evaluación a través de metaheurísticas y heurísticas como la ENS2 (*Extensive Neighborhood Search method*) y la regla de despacho EDD (*Earliest Due Date*). Para el caso de la heurística ENS2, se obtuvo que es uno de los mejores métodos en términos de efectividad basada en el intercambio e inserción de trabajos, ya que su tiempo computacional fue en promedio 8 segundos y su valor promedio de tardanza total fue de 2.24 unidades de trabajo. Así mismo, la regla de despacho EDD mostró un buen resultado en términos de tiempo computacional, siendo este en promedio menor a 0.5 segundos, pero un desempeño regular con respecto al valor promedio de tardanza total, el cual fue de 15.21 unidades de trabajo.

Framinan, Leisten, & Ruiz-Usano (2005) en su investigación compararon el desempeño de diferentes tipos de heurísticas para la minimización del tiempo total de terminación en un ambiente PFS determinístico. Los resultados obtenidos arrojaron que las heurísticas FL y B5FT (*Best 5 Flowtime*) son las sugeridas para resolver este tipo de problema, ya que la primera es mejor en términos de calidad de resultados con respecto a los valores obtenidos del tiempo total de terminación, mientras que la segunda es mejor en cuanto a menor tiempo computacional.

Ahora bien, los FFS estocásticos han sido estudiados en menor medida que los FS. Sin embargo, en la realidad, las industrias están sujetas al dinamismo del entorno y a diferentes factores que generan incertidumbre, por ejemplo, daños en las máquinas, tiempos de procesamiento variables, participación humana, condiciones ambientales, entre otros (Gonzalez-Neira, Ferone, Hatami, & Juan, 2017). Por tal motivo, surgen los modelamientos estocásticos, en los que los parámetros sujetos a incertidumbre se modelan a través de distribuciones de probabilidad. A pesar de que este comportamiento es el más cercano al contexto industrial, las investigaciones han sido pocas, comparado al modelo determinístico (Baker & Altheimer, 2012).

Uno de los supuestos más frecuentes en la literatura de *scheduling* es asumir que las máquinas están disponibles todo el tiempo, pero en la vida real, las máquinas están sujetas a daños y reparaciones (Fazayeli, Aleagha, Bashirzadeh, & Shafaei, 2016). Si bien la ocurrencia de los daños en las máquinas y su reparación podrían ser tratados desde el punto de vista determinístico, estos eventos, en realidad suceden de manera aleatoria.

De acuerdo con Gonzalez-Neira et al. (2017), el 80% de los trabajos investigados en FS y FFS estocásticos, se enfocan en problemas con un único objetivo. Adicionado a que un poco más de un 60% analizó solamente la minimización del valor esperado del *makespan*. No obstante, en la industria suelen tenerse varios objetivos que a menudo van en contravía, por lo tanto, el estudio de problemas multi-objetivo ha empezado a recibir más atención (Yenisey & Yagmahan, 2014). Adicionalmente, en la industria puede ser importante obtener soluciones que respondan de la mejor manera a los cambios dados por la incertidumbre, es decir, soluciones robustas. Dado que el enfoque robusto ha tratado tradicionalmente los parámetros bajo incertidumbre modelándolos con intervalos (Feng, Zheng, & Xu, 2016), los resultados probablemente no son tan precisos como si se buscara una solución robusta teniendo la distribución de probabilidad de los parámetros bajo incertidumbre. Por tanto, será importante mostrar con distintas distribuciones de probabilidad de los parámetros sujetos a aleatoriedad, que las soluciones son diferentes, y que por tanto un enfoque bajo intervalos no es el más adecuado.

Los autores Baker & Altheimer (2012) propusieron una metaheurística con la que generaron una solución a un problema de ambiente *flow shop* estocástico, en la que las tres heurísticas constructivas son capaces de obtener soluciones dentro del 1% de la optimalidad. Para ello eligieron tres distribuciones de probabilidad: exponencial, uniforme y lognormal, La distribución exponencial se especifica mediante un único parámetro (media) y corresponde al caso en el que la Heurística de Talwar produce un horario óptimo para dos máquinas. La distribución uniforme se especifica mediante dos parámetros (un valor medio y un rango) y permite influir en la medida en que las distribuciones de tiempo de procesamiento se superponen. Finalmente, la distribución lognormal es la base de la mayor parte del trabajo experimental, por dos razones. En primer lugar, lognormal también se puede especificar mediante dos parámetros (un valor medio y una desviación estándar). En segundo lugar, la distribución lognormal tiene un valor práctico considerable como modelo para los tiempos reales de procesamiento. Entre otros rasgos, una variante lognormal es no negativa, y sus parámetros pueden representar coeficientes de variación altos y bajos.

Debido a que la mayoría de los problemas FFS están clasificados como *NP-hard* (Ribas, Leisten, & Framiñan, 2010), es necesario realizar un enfoque metaheurístico para resolver instancias de mediano y gran tamaño en tiempos computacionales racionales (Ji, Yang, Duan, Wang, & Liu, 2016). Adicionalmente, a causa de la estocasticidad a tener en cuenta, se pretende trabajar en un marco metaheurístico con simulaciones (simheurística). Las simheurísticas son un enfoque flexible que se ha utilizado recientemente para resolver problemas combinatorios de optimización estocástica (SCOP por sus siglas en inglés) (Grasas, Juan, & Lourenço, 2014), donde se integran técnicas de simulación en un marco metaheurístico (Juan, Faulin, Grasman, Rabe, & Figueira, 2015). La metaheurística que se utilizará como método de optimización es GRASP, debido a que está basada en un procedimiento de búsqueda aleatorio y adaptativo, con un procedimiento multi-arranque, que construye su propia solución inicial (Feo & Resende, 1995) y que garantiza una buena solución a diferentes problemas de optimización. Esta metaheurística se hará hibridizada con simulación de Monte Carlo y la metodología *Pareto Archived Evolution Strategy* (PAES) propuesta originalmente por Knowles & Corne (2000) para tratar múltiples objetivos permitiendo hallar las fronteras de Pareto de estos.

Como aporte diferenciador, esta metaheurística se hará hibridizada con la simulación de Monte Carlo para resolver un problema estocástico que demuestre la variabilidad que se da en un ambiente de producción real a través de los daños que se pueden presentar en las máquinas y la metodología *Pareto Archived Evolution Strategy* (PAES) propuesta originalmente por Knowles & Corne (2000) para tratar múltiples objetivos permitiendo hallar las fronteras de Pareto de estos. Este caso de estudio busca dar solución a un problema FFS estocástico multi-objetivo, dado que, en la literatura la mayor parte de los estudios se enfocan en problemas FS determinísticos en los que se minimiza un único indicador, que por lo general es el *makespan*.

Con base en lo anterior, para obtener soluciones robustas se tendrán como funciones objetivo el valor esperado la tardanza total (*tardiness*), la desviación estándar de la tardanza, el valor esperado del tiempo total de terminación (*total completion time*) y la desviación estándar del *total completion time*. Para este proyecto, el tiempo total de terminación es igual al *flowtime*, debido a que los tiempos de liberación (*release times*) son iguales a cero (0). Los parámetros estocásticos elegidos serán los daños de las máquinas (*machine breakdowns*), los cuales contemplan tanto el tiempo de duración de los daños como el tiempo que sucede entre un daño y otro. Como resultado, se pretende mostrar a través de un diseño experimental, cómo afectan distintas distribuciones de probabilidad como la lognormal y uniforme frente a dos grados de variabilidad de los daños en las máquinas a las soluciones obtenidas.

De esta manera con la presente investigación se pretende dar respuesta a las siguientes preguntas:

- ¿Cómo es la Frontera de Pareto para los indicadores: valor esperado de la tardanza, desviación estándar de la tardanza, valor esperado del tiempo total de terminación y desviación estándar del tiempo total de terminación en un ambiente FFS con los daños de las máquinas como parámetros estocásticos?
- ¿Cómo se ven afectadas las soluciones al utilizar una simheurística bajo distintos grados de variabilidad, y distribuciones de probabilidad de los parámetros estocásticos?

2. Antecedentes

Antes de la existencia de la programación de la producción, en las fábricas era desconocido e incierto cuándo empezaba un trabajo, dónde estaba, cómo se movía en la planta y cómo debía ser realizado (Wiers, 1997). Hoy en día se han desarrollado diferentes enfoques y teorías para afrontar la programación de la producción, definida como la asignación de recursos disponibles a lo largo del tiempo para realizar un conjunto de tareas (Baker, 1974). Los problemas de programación son difíciles de resolver, debido a la naturaleza compleja de los problemas, por tal razón, en la mayoría de los estudios, los datos son determinísticos. Uno de estos problemas que ha sido ampliamente estudiados durante los últimos años, es el FS (Baker & Altheimer, 2012). A continuación, se presenta un resumen de las investigaciones realizadas en FS y FFS estocástico para posicionar la presente investigación respecto al estado del arte.

2.1 Ambiente Stochastic Flow shop (SFS)

Las primeras investigaciones sobre los modelos SFS, han sido ejecutadas por Banerjee (1965), quien ejecutó una secuenciación de una sola instalación con tiempos de ejecución aleatorios, y Makino (1965), quien desarrolló una regla de secuencia para encontrar una solución para dos trabajos en dos y tres máquinas, con el objetivo de minimizar el *makespan* esperado. La regla de Makino también puede aplicarse al caso de tres y cuatro trabajos en dos máquinas, si los tiempos de procesamiento se distribuyen exponencialmente.

Wang, Zhang, & Zheng (2005) proponen un algoritmo genético (GA por sus siglas en inglés) para resolver el problema de SFS, el cuál ha sido usado para resolver mayormente problemas determinísticos. El enfoque propuesto se basa en el rendimiento estadístico y una prueba de hipótesis, en búsqueda de minimizar el *makespan*. No sólo preserva la capacidad de búsqueda global de un GA, sino que también puede reducir las búsquedas repetidas de esas soluciones. Por otro lado, Baker & Altheimer (2012) plantean un método heurístico para resolver el problema SFS en el que se ejecutan tres procedimientos constructivos (valor modesto y predecible). Los principales experimentos consistieron en evaluar el problema en 450 casos en los que los tiempos de procesamiento se extrajeron de las distribuciones lognormales. Al comparar los rendimientos de estos procedimientos experimentales, se determina que logran un rendimiento casi óptimo. En el trabajo de Elyasi & Salmasi (2013a) minimizan el número esperado de trabajos tardíos en un problema de programación dinámica de m máquinas en un ambiente FS. Los tiempos de procesamiento de los trabajos son determinísticos, pero las fechas de vencimiento son estocásticas, con llegadas aleatorias a la estación. Emplean un método dinámico para este problema, descomponiéndolo en m subproblemas estocásticos de una sola máquina.

Rahmani, Ramezani, & Mehrabad (2014) proponen un modelo matemático para la solución de un problema multi-objetivo en un ambiente SFS con el fin de minimizar el tiempo total de terminación, la tardanza

total y el *makespan*, teniendo como parámetros estocásticos los tiempos de procesamiento y las fechas de liberación. Los enfoques utilizados para tratar el problema multi-objetivo y los parámetros estocásticos fueron la implementación de *fuzzy goal programming* (FGP) y *chance constrained programming* (CCO) respectivamente.

Por otra parte, Ferone, Gruler, Festa, & Juan (2016) exponen una solución para el problema PFS estocástico basado en una simheurística que combina la simulación de Monte Carlo con la metaheurística GRASP con tiempos de procesamiento aleatorios con el objetivo de minimizar el tiempo total de terminación y el *makespan*. Dado los buenos resultados obtenidos, resaltan la importancia del uso de las simheurísticas para tratar problemas de optimización combinatorios con aspectos estocásticos complejos, a gran escala y que frecuentemente se pueden aplicar en problemas logísticos y de transporte a nivel industrial. Del mismo modo, Juan, Barrios, Vallada, Riera, & Jorba (2014) desarrollaron una simheurística que involucraba la simulación de Monte Carlo y la metaheurística *iterated local search* (ILS) para tratar con el comportamiento estocástico del problema SPFS. Esta investigación se basó en minimizar el *makespan* tanto para la versión determinística como para la versión estocástica. Los resultados arrojaron que la metodología permite resolver este problema con cualquier tipo de distribución y que, haciendo uso de un algoritmo metaheurístico eficiente, se puede generar un conjunto reducido de soluciones de alta calidad.

2.2 Ambiente Stochastic Flexible Flow Shop (SFFS)

En años recientes se han realizado diferentes estudios con parámetros estocásticos. Almeder & Hartl (2013) plantean una solución para un caso real de un problema multi-objetivo en un ambiente SFFS que cuenta con una capacidad limitada de *buffers*, con el fin de minimizar cuatro diferentes criterios enfocados en evitar la congestión del sistema, los cuales son: *average expected utilization of machine 1*, *average expected utilization of the buffer*, *average expected completion (or throughput) time* y la combinación de los tres objetivos mencionados anteriormente donde el primero representa el 50% y los dos siguientes representan cada uno el 25% del objetivo en general. Para su solución utilizaron la metaheurística *Variable Neighborhood Search* (VNS) y, para tratar el ámbito multi-objetivo el enfoque se basó en la comparación entre dos criterios de aceptación: *t-test* y *sample of scenarios* (SAE).

Lin, Huang, & Lin (2016) determinan apropiado considerar el número de máquinas en funcionamiento por etapa como un parámetro estocástico. Para ello, los autores modelaron el problema como un *multistate flexible flow shop network* (MFFSN), donde cada arco denota una estación con capacidad estocástica y cada nodo denota un *buffer*. El objetivo de esta investigación es maximizar la probabilidad de que el MFFSN pueda completar el pedido de un cliente, compuesto por múltiples tipos de trabajos dentro de un umbral de tiempo, para lo cual, se propone un algoritmo *branch-and-bound*, para esta ejecución, se presentan dos casos prácticos: un sistema de producción de baldosas y un sistema de ropa.

La cantidad de trabajos de SFFS es menor a la encontrada en SFS. Aun así, se puede evidenciar que existen investigaciones que estudian el problema SFFS como la propuesta por los autores Yang, Kuo, & Cho (2007) en donde minimizan el tiempo total de tardanza, mediante un enfoque de simulación de algoritmos genéticos (GA), para resolver un problema de decisión de despacho combinatorio (MACD), cuyos parámetros son los tiempos de configuración, tiempo medio entre fallas (MTBF) y tiempo medio de reparación (MTTR). Gholami, Zandieh, & Alem-Tabriz (2009) afrontan la manera de incorporar la simulación en el enfoque de algoritmo genético de clave aleatoria (RKGGA) con máquinas que sufren una avería estocástica, para minimizar el *makespan* con un enfoque *sequence dependent setup times* (SDST) en un ambiente SFFS. Dugardin, Yalaoui, & Amodeo (2010) se enfocan en la maximización de la tasa de utilización del cuello de botella y la minimización del tiempo máximo de finalización, mediante un algoritmo genético (GA) multi-objetivo llamado L-NSGA que utiliza la relación de dominio de Lorenz y propone como parámetro estocástico los tiempos de procesamiento; Mirabi, Ghomi, & Jolai (2011) consideran un FFS de dos etapas en la condición de avería de máquina, es decir, que no está siempre disponible durante el periodo de programación, cuyo objetivo es encontrar las combinaciones de trabajo óptimas y el horario de trabajo óptimo, de modo que el *makespan* se minimice. Para resolver este problema, se introduce un enfoque óptimo para la precedencia de trabajo cuando hay una máquina en ambas etapas y un algoritmo heurístico cuando hay m máquinas en la etapa dos; Choi & Wang (2012) presentan una

visión basada en la descomposición (DBA), que combina el tiempo de procesamiento más corto (SPT) y el algoritmo genético (GA) para minimizar el *makespan* de un (FFS) con tiempos de procesamiento estocásticos.

Con base en lo anterior, se observa que las investigaciones en torno a los problemas de FFS y especialmente con metodología de resolución GRASP son pocos. Por lo tanto, con la metodología GRASP, se probarán diferentes instancias bajo variaciones de las distribuciones de probabilidad y coeficientes de variación de los tiempos de los daños en las máquinas y el tiempo entre daños, para determinar su efecto en las cuatro funciones objetivo, y mostrar así la importancia de hacer un ajuste preciso de las distribuciones de los parámetros estocásticos.

3. Objetivos

General:

Diseñar una metaheurística multi-objetivo que integre GRASP con la simulación de Monte Carlo y la metodología PAES para solucionar el problema FFS que considera los tiempos de duración de los daños de las máquinas y los tiempos entre daños como parámetros estocásticos, con el fin de minimizar el valor esperado de la tardanza total, la desviación estándar de la tardanza, el valor esperado del tiempo total de terminación y la desviación estándar de la tardanza.

Específicos:

- Desarrollar una simheurística que minimice el valor esperado de la tardanza total, la desviación estándar de la tardanza, el valor esperado del tiempo total de terminación y la desviación estándar del tiempo total de terminación en un ambiente FFS, para construir la frontera de Pareto de dichos indicadores en el ambiente mencionado.
- Comparar el desempeño del algoritmo propuesto para el caso determinístico contra las reglas de despacho ENS2 y FL que han probado buenos resultados en FS y *permutation flow shop* (PFS) con minimización de tardanza total y del tiempo total de terminación respectivamente.
- Realizar un diseño experimental con el fin de evaluar cómo afectan los distintos grados de variabilidad y distribuciones de probabilidad de los daños estocásticos en las máquinas en los objetivos a minimizar.

4. Metodología

4.1 Metaheurística GRASP

Los autores Feo & Resende (1995) son los pioneros en desarrollar la metaheurística GRASP, la cual es una técnica de muestreo aleatorio iterativo en el que cada iteración proporciona una solución al problema en cuestión. Ésta consiste en dos fases: fase de construcción y fase de búsqueda local. En la primera, se construye una solución inicial a través de una función de utilidad *greedy*, es decir, en cada paso de esta primera etapa, la selección del siguiente trabajo a agregar a la secuencia es determinada por medio del ordenamiento de todos los elementos candidatos en una lista de candidatos restringida (RCL por sus siglas en inglés) con respecto a la función de utilidad definida. En la segunda fase, se aplica un algoritmo de búsqueda local a la solución construida con el fin de encontrar una mejora o una solución localmente óptima. La búsqueda local funciona de forma iterativa reemplazando sucesivamente la solución actual por una mejor solución de su vecindario y termina cuando no se encuentra una mejor solución en dicho vecindario.

4.1.1 Notación

Para mayor entendimiento de los pseudocódigos que se presentan para cada una de las fases de la metaheurística GRASP, se presenta primero en la Tabla 1 la descripción de parámetros y variables utilizados:

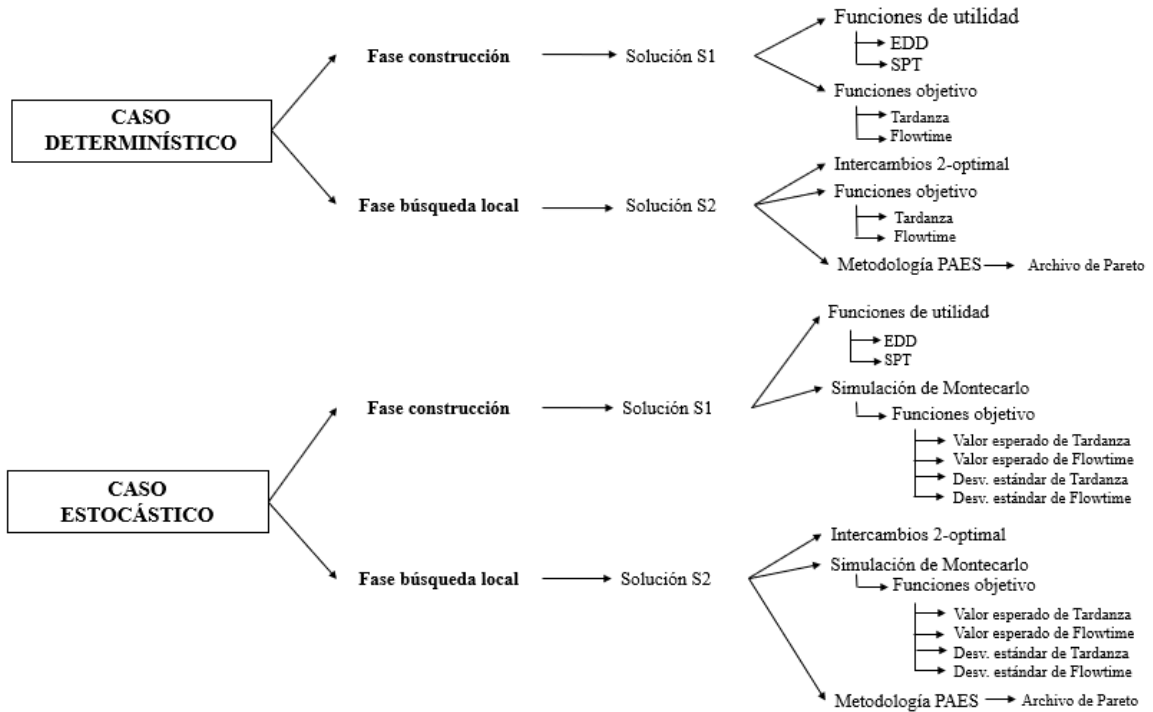
Tabla 1. Notación metaheurística GRASP

| Notación | Significado |
|--------------------|---|
| I | Número de iteraciones |
| N | Número de trabajos |
| E | Número de estaciones |
| M | Número de máquinas |
| it | Índice para las iteraciones |
| j | Índice para los trabajos |
| i | Índice para las máquinas |
| k | Índice para las estaciones |
| g | Índice para el número de GRASP |
| $S1$ | Secuencia Fase de construcción |
| $S2$ | Secuencia Fase de búsqueda local |
| NS_j | Vector de trabajos $j \in N$ no secuenciados |
| S_{min} | Menor valor para la FU |
| S_{max} | Mayor valor para la FU |
| α | Parámetro codicioso |
| FU | Función de utilidad |
| RCL | Lista restringida de candidatos |
| d_j | Fecha de entrega del trabajo $j \setminus j \in N$ |
| S_{jk} | Tiempo de inicio del trabajo j en la estación $k \setminus j \in N, k \in E$ |
| C_{jk} | Tiempo de finalización del trabajo j en la estación $k \setminus j \in N, k \in E$ |
| FT | Tiempo total de terminación (<i>flowtime</i>) |
| T | Tardanza |
| FO | Función objetivo |
| VE | Valor esperado |
| DE | Desviación estándar |
| Ag | Porcentaje de las máquinas que presentan daño |
| $MTTR$ | Tiempo promedio que toma reparar una falla |
| $MTBF$ | Tiempo promedio en el que el equipo no falla |
| CV | Coefficiente de variación |
| t | Índice de tiempo |
| d | Índice para los daños |
| $dañoS_{dik}$ | Tiempo de inicio del daño d , en la máquina i de la estación $k \setminus i \in M, k \in E$ |
| $dañoF_{dik}$ | Tiempo de finalización del daño d , en la máquina i de la estación $k \setminus i \in M, k \in E$ |
| $dañosM_{ik}$ | Número de daños en la máquina i de la estación $k \setminus i \in M, k \in E$ |
| $SeqEMJ_{k i pos}$ | Trabajo secuenciado en la máquina i de la estación k en el momento $pos \setminus i \in M, k \in E$ |
| L_j | Retardo del trabajo $j \setminus j \in N$ |
| T_j | Tardanza del trabajo $j \setminus j \in N$ |

4.1.2 Metodología de solución

Para el desarrollo de este proyecto, inicialmente fue necesario evaluar el desempeño de la metaheurística GRASP en el escenario determinístico, con el fin de proceder a la implementación de la simheurística para el caso estocástico. En la Ilustración 1 se presenta el esquema de la metodología de solución para este caso de estudio.

Ilustración 1: Esquema general de la metodología de solución

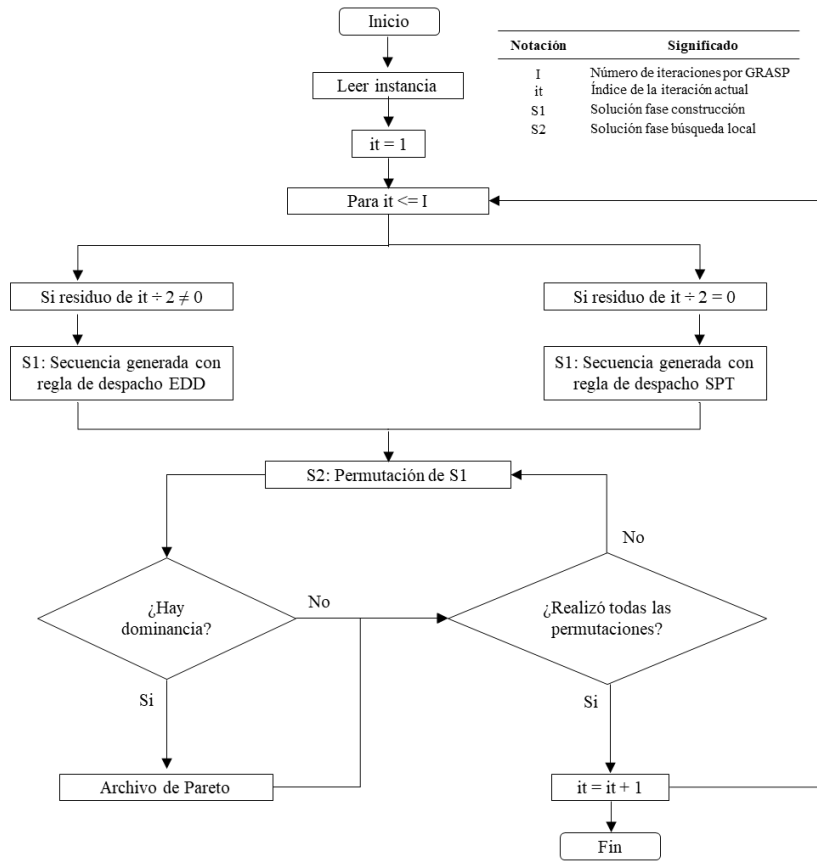


4.1.3 Metaheurística GRASP para la solución del problema de determinístico

Así pues, se ejecuta el método asumiendo el escenario determinístico a partir de las 324 instancias dadas por los autores (Pan, Ruiz, & Alfaro-Fernández, 2017). Cabe resaltar que dado que las instancias *benchmark* son para un FFS con ventanas de tiempo para la entrega, se tomó como fecha de entrega de los trabajos (*due date*) el límite superior del *due date window* de cada instancia.

De acuerdo con lo mencionado anteriormente, se presenta un diagrama de flujo (Ilustración 2) en donde se resumen los pasos de la metaheurística GRASP para el caso determinístico.

Ilustración 2. Diagrama de flujo de la metaheurística GRASP escenario determinístico



Nota: La evaluación de desempeño de la metaheurística se puede encontrar en el numeral 6 de resultados.

4.1.2.1 Fase de construcción

La fase de construcción requiere de una función *greedy* que debe ser definida dependiendo de la función objetivo a minimizar para obtener una buena solución inicial. En cada iteración de esta etapa, sólo un elemento, en este caso un trabajo, se agrega a la solución dependiendo de su función *greedy*. El trabajo seleccionado para la solución se toma aleatoriamente de un subconjunto de trabajos no programados que tienen mejores valores $\alpha\%$ de la función de costo. Este subconjunto se llama lista de candidatos restringida (RCL por sus siglas en inglés) (González-Neira & Montoya-Torres, 2017). El parámetro codicioso (α) es determinado experimentalmente y su valor varía de cero a uno. Cuando el valor de α es cero, la metaheurística es totalmente codiciosa y cuando es 1, es totalmente aleatoria (Feo & Resende, 1995).

Para los dos casos de estudio: determinístico y estocástico, se tomaron como funciones de utilidad la adaptación de las reglas de despacho SPT (*Shortest Processing Time*) y EDD (*Earliest Due Date*) al problema FFS, las cuales contribuyen a la minimización del tiempo total de terminación y la tardanza total respectivamente. La regla de despacho SPT, definida originalmente para el problema de una máquina, consiste en seleccionar el trabajo con el menor tiempo de procesamiento (P_j) (Brah & Wheeler, 1998). En el caso del FFS, el tiempo total de procesamiento de un trabajo no es simplemente la suma de los tiempos de procesamiento del mismo en cada una de las estaciones, sino es la diferencia entre el tiempo de terminación en la última estación y el tiempo de inicio de procesamiento del trabajo en la primera estación (1). Este tiempo se actualiza cada vez que se secuencia un trabajo a partir del tiempo de disponibilidad de las máquinas.

$$P_j = C_{j|k|} - S_{j1} \quad (1)$$

El pseudocódigo de la fase de construcción con la regla de despacho SPT se muestra en el Pseudocódigo 1.

Pseudocódigo 1: Pseudocódigo fase de construcción - función de utilidad SPT

```

1. Inicializar candidatos del vector NS
2. j=1
3. Hacer mientras j ≤ N
    4. Si NSj = 1 entonces
        5. Smin = Min {Cj|k| - Sj1 \ j ∈ N}
        6. Smax = Max {Cj|k| - Sj1 \ j ∈ N}
        7. FU = Smin + α (Smax - Smin)
        8. RCL = {Cj|k| - Sj|k| ≤ FU }
        9. Seleccionar aleatoriamente un trabajo de la RCL
        10. Actualizar candidato en vector NSj = 0
        11. j=j+1
12. Fin mientras
13. Calcular FT y T para S1
14. Fin Fase de construcción

```

Por otro lado, la regla de despacho EDD, originalmente propuesta también para el problema de una máquina, secuencia los trabajos ascendente según sus fechas de entrega (d_j). Como las fechas de entrega no se ven afectadas por el tipo de ambiente, para el FFS la función de utilidad sigue manteniendo el ordenamiento por los d_j . Una vez que una máquina está disponible, se escoge aleatoriamente un trabajo de la RCL y éste es asignado para su procesamiento (González-Neira & Montoya-Torres, 2017). El pseudocódigo de la fase de construcción con la regla de despacho EDD se presenta en el Pseudocódigo 2.

Pseudocódigo 2: Pseudocódigo fase de construcción – función de utilidad EDD

```

1. Inicializar candidatos del vector NS
2. j=1
3. Hacer mientras j ≤ N
    4. Si NSj = 1 entonces
        5. Smin = Min {dj \ j ∈ N}
        6. Smax = Max {dj \ j ∈ N}
        7. FU = Smin + α (Smax - Smin)
        8. RCL = {dj ≤ FU }
        9. Seleccionar aleatoriamente un trabajo de la RCL
        10. Actualizar candidato en vector NSj = 0
        11. j=j+1
12. Fin mientras
13. Calcular FT y T para S1
14. Fin Fase de construcción

```

Ahora bien, en cada iteración de la metaheurística se alternan las dos funciones de utilidad según el número correspondiente a cada iteración, es decir, cuando una iteración es impar el GRASP se ejecuta según la regla de despacho EDD y cuando una iteración es par, el GRASP se ejecuta según la regla de despacho SPT.

4.1.2.2 Fase de búsqueda local

Para la fase de búsqueda local se seleccionaron los intercambios *2-optimal* originalmente propuestos por Croes (1958) para el problema del agente viajero. Para el caso del problema de programación de la producción, el *2-optimal* busca intercambiar dos trabajos de la secuencia S1 de la fase de construcción para así comparar el valor de la función objetivo anterior contra la función objetivo actual de la secuencia S2 de la fase de búsqueda local. Este proceso tiene dos criterios de parada: se realiza hasta haber intercambiado todos los trabajos y hasta que se haya obtenido el mejor resultado posible de la función objetivo.

Debido a que, para el escenario determinístico se busca minimizar la tardanza total y el tiempo total de terminación, se procede a implementar un algoritmo evolutivo que permita la optimización de un problema multi-objetivo llamado algoritmo PAES, el cual fue desarrollado inicialmente como un método de búsqueda local multi-objetivo para encontrar soluciones al problema de ruteo fuera de línea (Knowles & Corne, 1999). En su forma base el PAES emplea la búsqueda local para la generación de nuevas soluciones candidatas, pero utiliza la información de la población para ayudar al cálculo de la solución. El algoritmo tiene tres formas de ejecutarse: (1 + 1)-PAES, (1 + λ)-PAES, (μ + λ)-PAES. La primera forma (1 + 1)-PAES está compuesto por tres partes: el generador de la solución candidata, la función de aceptación de la solución candidata y el archivo de soluciones no dominadas (NDS *archive* por sus siglas en inglés). El generador de la solución candidata mantiene una única solución actual y en cada iteración se produce una única mutación aleatoria candidata (Knowles & Corne, 2000). La segunda forma (1 + λ)-PAES es muy similar a la primera, excepto que los mutantes λ se generan a partir de la solución actual. Cada uno se compara con esta última a través del archivo y el mejor de estos reemplaza la solución actual. La tercera forma (μ + λ)-PAES mantiene una población de tamaño μ a partir de la cual se generan λ copias, seleccionando aquellas que tengan mejor *fitness* con respecto a las del archivo (Knowles & Corne, 1999).

Para este caso de estudio, se aplicó el algoritmo (1 + 1)-PAES, puesto que su rendimiento es fuerte especialmente dada su baja complejidad computacional, incluso en tareas exigentes donde se podría esperar que los métodos de búsqueda local estén en desventaja (Knowles & Corne, 2000). Adicionalmente, se adaptó el *test* propuesto por estos autores para determinar si es relevante aceptar o rechazar una solución dentro del archivo, teniendo en cuenta que éste tiene capacidad infinita.

Teniendo en cuenta lo anterior, en el Pseudocódigo 3 se presenta el pseudocódigo de la fase de búsqueda local de la metaheurística GRASP y la adaptación de la metodología PAES propuesta por los autores (Knowles & Corne, 2000).

Pseudocódigo 3: Pseudocódigo fase de búsqueda local de la Metaheurística GRASP y la adaptación de la metodología (1+1 - PAES)

```

1. Leer S1 de la Fase de construcción
2. con1=0
3. Mientras con1 < N
4.   cont2=cont1+1
5.   Mientras con2 < N+1
6.     S2 = S1
7.     Modificar S2 cambiando trabajos de posición con1 y con2
9.     Calcular funciones objetivo  $T_{S2}$  y  $FT_{S2}$ 
10.    Si  $T_{S2} < T_{S1}$  entonces
11.      Si  $FT_{S2} \leq FT_{S1}$  entonces
12.        S1=S2
13.        con1 = 0
14.        con2 = cont1 + 1
15.        Eliminar secuencias menores a S2 que estén en el archivo
16.      Si no
17.        con2=con2+1
18.        Agregar S2 al archivo
19.        Eliminar las secuencias menores a S2 que estén en el archivo
20.        Si S2 está en la región menos poblada del archivo entonces
21.          S1=S2
22.          con1 = 0
23.          con2 = cont1 + 1
24.        Fin si
25.    Fin si
26.  Si no
27.    Si  $T_{S2} = T_{S1}$  entonces
28.      Si  $FT_{S2} < FT_{S1}$  entonces
29.        S1=S2
30.        con1 = 0
31.        con2 = cont1 + 1

```

```

32. Eliminar secuencias menores a S2 que estén en el archivo
33. Si no
34. con2= con2 +1
35. S2 no entra en el archivo
36. Fin si
37. Si no
38. Si  $FT_{S2} = FT_{S1}$  entonces
39. con2=con2+1
40. Agregar S2 al archivo
41. Eliminar las secuencias menores a S2 que estén en el archivo
42. Si S2 está en la región menos poblada del archivo entonces
43. S1=S2
44.con1 = 0
45.con2 = cont1 + 1
46.Fin si
47. Si no
48. con2= con2 +1
49. S2 no entra en el archivo
50. Fin si
51. Fin si
52. Fin si
53. Fin mientras
54.Fin mientras
55.con1 = con1+1
56. Fin fase de búsqueda local

```

4.1.3 Simheurística para la solución del problema multi-objetivo en el caso estocástico

Conservando lo mencionado para el escenario determinístico, la fase de construcción consta de dos reglas de despacho: SPT y EDD como funciones de utilidad. Sin embargo, para el caso estocástico la simheurística comprende la generación, a través de números aleatorios, de los tiempos de duración de los daños de las máquinas (MTTR por sus siglas en inglés) y tiempos entre fallas (MTBF por sus siglas en inglés), dadas unas distribuciones de probabilidad y coeficientes de variación. Adicionalmente, la simheurística involucra el procedimiento de simulación de Monte Carlo para hallar las cuatro funciones objetivo: valor esperado del *flowtime*, valor esperado de la tardanza, desviación estándar del *flowtime* y desviación estándar de la tardanza.

Así, la simheurística GRASP es la integración de la metaheurística GRASP propuesta para el caso determinístico (ver Ilustración 2) con la simulación de Monte Carlo, es decir, cada vez que se obtiene una solución, ésta debe ser simulada. Las distribuciones de probabilidad seleccionadas tanto para el MTTR como para el MTBF fueron uniforme y lognormal. Para establecer la media de ambos tiempos se utilizó el procedimiento planteado por los autores (Gholami, Zandieh, & Alem-Tabriz, 2009). Para el MTTR se escogieron como valores esperados $0,1\bar{P}$ y \bar{P} , donde \bar{P} es el tiempo total promedio de procesamiento por trabajo. Por otra parte, los valores esperados del MTBF se seleccionaron por medio del uso del porcentaje de tiempo en el que las máquinas presentan daños (Ag). Esta medida se calcula como indica la ecuación (2).

$$Ag = \frac{MTTR}{MTTR + MTBF} \quad (2)$$

A partir de los valores de Ag y MTTR, se procede a calcular el valor del MTBF (3):

$$MTBF = \frac{MTTR}{Ag} - MTTR \quad (3)$$

Los valores de Ag utilizados para este diseño experimental fueron 5% y 15% y fueron elegidos a partir del intervalo propuesto por (Gholami et al., 2009) el cual varía entre $(0.026 \% \leq Ag \leq 20.83)$.

El pseudocódigo para programar los daños se presenta en el Pseudocódigo 4.

Pseudocódigo 4: Pseudocódigo schedule de daños

```

1. Para k = 1 hasta E
2. Para i = 1 hasta M
3. t = 1
4. d = 1
5. Mientras j= 1 hasta Cantidad de trabajos de cada estación en cada máquina
6. Escoger el trabajo a programar
7. Mientras el trabajo no haya sido programado hacer
8. Si k > 1 entonces
9. Si  $C_{j\ k-1} \geq t$  entonces
10.  $t = C_{j\ k-1} + 1$ 
11. Fin si
12. Fin si
13. Si  $d \leq \text{daños}M_{ik}$  entonces
14. Si d > 1 entonces
15. Si  $t \geq \text{daño}F_{d-1,i,k}$  y  $t + P_{(\text{trabajo}, k)} \leq \text{daño}S_{dik}$  entonces
16.  $S_{(\text{trabajo}, k)} = t$ 
17.  $t = S_{(\text{trabajo}, k)} + P_{(\text{trabajo}, k)}$ 
18.  $C_{(\text{trabajo}, k)} = t$ 
19. Actualizar el trabajo como programado
20. Si no
21.  $t = \text{daño}F_{(d,i,k)} + 1$ 
22. d= d+1
23. fin si
24. si no
25. Si  $t \geq 1$  y  $t + P_{(\text{trabajo}, k)} \leq \text{daño}S_{dik}$  entonces
26.  $S_{(\text{trabajo}, k)} = t$ 
27.  $t = S_{(\text{trabajo}, k)} + P_{(\text{trabajo}, k)}$ 
28.  $C_{(\text{trabajo}, k)} = t$ 
29. Actualizar el trabajo como programado
30. Si no
31.  $t = \text{daño}F_{(d,i,k)} + 1$ 
32. d= d+1
33. fin si
34. Fin si
35. si no
36.  $S_{(\text{trabajo}, k)} = t$ 
37.  $t = S_{(\text{trabajo}, k)} + P_{(\text{trabajo}, k)}$ 
38.  $C_{(\text{trabajo}, k)} = t$ 
39. Actualizar el trabajo como programado
40. Fin si
41. Fin mientras
42. Fin mientras
43. Fin i
44. Fin k

```

En segunda instancia, se definieron arbitrariamente dos coeficientes de variación (0.2 y 0.4) para hallar la desviación estándar ($\sigma = cv \times \mu$) del MTBF y del MTTR. Considerando que por el teorema del límite central con una muestra de $n \geq 30$ la variable aleatoria media muestral (\bar{X}) se distribuye normal, se determinó un número de 100 corridas para cada simulación de Monte Carlo (ver Pseudocódigo 5), de tal manera que se pudiera calcular para cada secuencia obtenida el valor de las cuatro funciones-objetivo propuestas.

Pseudocódigo 5: Pseudocódigo de la simulación de Monte Carlo

```

1. Leer parámetros de entrada ( $\bar{P}$ , Ag, ~ MTTR, CV MTTR, ~ MTBF, CV MTBF)
2. Para corrida = 1 hasta 100
3. Para k = 1 hasta E

```

```

4. Para i = 1 hasta M
    5. t = 1
    6. d = 1
    7. Hacer mientras t < TiempoTotal / E
        8. Generar MTTR
        9. Generar MTBF
        10.  $dañoS_{dik} = t + MTBF$ 
        11.  $dañoF_{dik} = dañoS_{dik} + MTTR$ 
        12.  $t = dañoF_{dik}$ 
        13. d = d + 1
    14. Fin mientras
    15.  $dañosM_{ik} = d$ 
16. Fin i
17. Fin k
18. Programar los daños
19. Calcular FT y T
20. Fin corrida
21. Calcular  $\mu_{FT}, \sigma_{FT}, \mu_T, \sigma_T$ .

```

En tercera instancia, para tratar conjuntamente el problema multi-objetivo se hace uso también de la metodología (1+1)-PAES, pero realizando una adaptación a ésta al momento de construir la frontera de Pareto para las cuatro funciones objetivo, corroborando la dominancia existente entre la secuencia S1 de la fase de construcción y la secuencia S2 de la fase de búsqueda local.

4.2 Reglas de despacho utilizadas para evaluar el desempeño de la metaheurística GRASP (caso determinístico)

Como parte de la investigación se realizó una comparación de la metaheurística GRASP contra las reglas de despacho ENS2 y FL, con el propósito de analizar el desempeño del algoritmo propuesto para la minimización de las funciones objetivo de la tardanza total y el tiempo total de terminación en el escenario determinístico. Ambas reglas de despacho son adaptaciones de la heurística NEH, la cual fue presentada inicialmente por los autores Nawaz, Enscore, & Ham (1983) y que consta de dos fases: primero, los trabajos se organizan de manera descendente con respecto a la suma de sus tiempos de procesamiento. Dentro de la segunda fase, se construye una secuencia de trabajo evaluando los *schedules* parciales que se originan en la fase inicial (Framinan, Leisten, & Ruiz-Usano, 2002).

La regla de despacho FL es una heurística constructiva planteada por los autores (Framinan & Leisten, 2003), la cual probó buenos resultados computacionales en términos de calidad de las soluciones y número promedio de veces que alcanza la solución óptima. Ésta consiste originalmente en seleccionar cada trabajo y evaluar su tiempo de terminación promedio, a partir de la suma de sus tiempos de procesamiento en todas las estaciones. Seguido de esto, se ordenan los trabajos de manera ascendente para seleccionar el que mejor tiempo de terminación parcial tenga. En las siguientes iteraciones, se inserta el siguiente trabajo en cada posible posición de la secuencia actual y se selecciona la secuencia en la que se obtenga el menor tiempo de terminación. El paso a paso de la regla de despacho FL se presenta en el Pseudocódigo 6.

Pseudocódigo 6: Regla de despacho FL propuesta por los autores (Framinan & Leisten, 2003)

```

1. Leer datos
2. Generar secuencia de trabajos ascendentemente de acuerdo con el  $P_j$ 
3. Elegir el primer trabajo con menor  $P_j$  para iniciar la secuencia
4. Ubicar el siguiente trabajo con mejor  $P_j$  a la secuencia y realizar intercambios entre cada par de trabajos en la secuencia dada, es decir, cambiar la posición de los trabajos colocados en las posiciones i-ésima y j-ésima para todos los i y j.
5. Calcular el Flowtime para las secuencias generadas en el paso 4.
    $Flowtime = C_{j|k|} \setminus j$  es el último trabajo secuenciado
6. Elegir la secuencia con mejor Flowtime.
7. Volver al paso 4 hasta que se hayan secuenciados todos los N trabajos.
8. Retorna la secuencia con mejor Flowtime.

```

9. Fin regla de despacho FL.

La regla de despacho ENS2 presentada por (Kim, 1993; Kim, Lim, & Park, 1996) es un algoritmo que comienza a partir de una secuencia completa inicial NEHed, en donde los trabajos son ordenados ascendentemente según sus fechas de entrega y que en cada iteración trata de mejorar la secuencia moviéndose a un mejor vecino de la misma, por lo tanto, es llamado método de mejora iterativa. Una secuencia vecina se obtiene a través del intercambio de un par de trabajos de la secuencia inicial, es decir, intercambiando trabajos ubicados en la i -ésima y j -ésima posición para todos los trabajos. En el Pseudocódigo 7 se muestra el paso a paso de la regla de despacho ENS2.

Pseudocódigo 7: Paso a paso regla de despacho ENS2

1. Leer datos
2. Generar secuencia de trabajos ascendentemente de acuerdo con el d_j
3. Elegir el primer trabajo con menor d_j para iniciar la secuencia
4. Ubicar el siguiente trabajo con mejor d_j en todas las posibles posiciones de la secuencia.
5. Calcular la tardanza (T_j) para las posibles secuencias.
$$L_j = C_j - d_j$$
$$T_j = \text{Max} \{0, L_j\}$$
6. Elegir la secuencia con mejor tardanza.
7. Volver al paso 4 hasta que se hayan secuenciados todos los N trabajos
8. Realizar intercambios entre cada par de trabajos en la secuencia dada, es decir, cambiar la posición de los trabajos colocados en las posiciones i -ésima y j -ésima para todos los i y j . (Y.-D. Kim et al., 1996).
9. Elegir la secuencia con mejor tardanza
10. Fin regla de despacho ENS2.

5. Componente de Diseño en ingeniería

5.1 Declaración de Diseño

El diseño principal de esta propuesta es una simheurística que dé soluciones al problema de la programación de la producción en un ambiente FFS estocástico donde los tiempos MTTR y MTBF son variables aleatorias. El problema abordado considera múltiples objetivos, donde las desviaciones estándar se incluyen para la obtención de soluciones robustas. La simheurística involucrará la metaheurística GRASP para la optimización del problema y la metodología PAES para el tratamiento de los múltiples objetivos.

5.2 Proceso de Diseño

El diseño de la metaheurística GRASP está compuesto por dos etapas: la primera etapa se basa en el desarrollo y programación de la misma para la solución del problema determinístico, teniendo en cuenta la metodología PAES. La segunda etapa consiste en hibridizar la metaheurística con la simulación de Monte Carlo (simheurística) y adaptando la metodología PAES para la solución del problema estocástico multi-objetivo.

La programación de la metaheurística GRASP para el caso determinístico se realizó a través de la plataforma *Visual Basic for Applications* (VBA). Esta etapa fue necesaria para comprobar su efectividad frente a las reglas de despacho, para así, proceder al desarrollo de la segunda etapa. Para lograr esto, se corrieron 324 instancias de 10, 15 y 20 trabajos, tanto para la metodología propuesta como para las reglas de despacho, logrando obtener buenos resultados. La programación de la simheurística para el caso estocástico se desarrolló de igual manera en la plataforma VBA y se corrieron las mismas 324 instancias, considerando las dos distribuciones de probabilidad con sus dos coeficientes de variación tanto para el MTBF como para el MTTR.

A continuación, se muestra la Tabla 2 en donde se presentan las características de los equipos utilizados para correr las instancias para cada uno de los casos:

Tabla 2: Información equipos

| Marca | Cantidad | Procesador | Memoria RAM | Sistema Operativo |
|-------|----------|---------------------------------|-------------|-------------------|
| HP | 1 | Intel(R) Core(TM) i5 - 2.60GHz | 8 GB | Windows 10 Pro |
| ASUS | 1 | Intel(R) Core(TM) i5 - 2.40 GHz | 8 GB | Windows 10 Pro |
| SONY | 1 | Intel(R) Core(TM) i5 - 2.50GHz | 4 GB | Windows 10 Home |
| HP | 40 | Intel(R) Core(TM) i5 - 3.50 GHz | 8 GB | Windows 10 Pro |

Por otro lado, en la Tabla 3 se muestran los tiempos de corrida promedio para cada caso estudiado según el tipo de instancia evaluada.

Tabla 3: Tiempos de corrida promedio para el caso determinístico y estocástico

| N | M | E | TIEMPO PROMEDIO DETERMINÍSTICO (MINUTOS) | TIEMPO PROMEDIO ESTOCÁSTICO (MINUTOS) |
|----|----|---|--|---------------------------------------|
| | | | | |
| 10 | 4 | 2 | 0,039 | 187,2 |
| | 6 | 2 | 0,049 | 200,64 |
| | 6 | 3 | 0,052 | 302,4 |
| | 9 | 3 | 0,054 | 316,8 |
| | 8 | 4 | 0,073 | 408 |
| | 12 | 4 | 0,093 | 475,2 |
| 15 | 4 | 2 | 0,049 | 609,6 |
| | 6 | 2 | 0,057 | 652,8 |
| | 6 | 3 | 0,069 | 1118,4 |
| | 9 | 3 | 0,075 | 1204,8 |
| | 8 | 4 | 0,084 | 1694,4 |
| | 12 | 4 | 0,102 | 1752 |
| 20 | 4 | 2 | 0,139 | 782,4 |
| | 6 | 2 | 0,154 | 916,8 |
| | 6 | 3 | 0,196 | 1516,8 |
| | 9 | 3 | 0,209 | 1608 |
| | 8 | 4 | 0,448 | 2428,8 |
| | 12 | 4 | 0,462 | 2457,6 |

5.3 Requerimientos de desempeño

Para el modelamiento de los daños en las máquinas se seguirán los parámetros establecidos en la norma ISO 28640 para la generación de números aleatorios como si fueran variables aleatorias reales. Todo lo anterior será tomado en cuenta en la programación en *Visual Basic for Applications* (VBA).

De igual forma, se debe garantizar lo siguiente:

- La simheurística debe generar soluciones factibles, donde se incluyan todos los trabajos a programar, se obtengan los valores esperados de las funciones objetivo, bajo distintas distribuciones de probabilidad y grados de variabilidad de los daños en las máquinas, y como resultado, generar unas fronteras de Pareto para cada una de ellas.
- La simheurística debe correr sin tener que alterar el código, adaptándose fácilmente a cualquier cambio de los parámetros.

5.4 Pruebas de rendimiento

Con el fin de evaluar el desempeño de la metodología propuesta y de garantizar que cumpla con los requerimientos planteados, se realizaron diferentes diseños experimentales, cuyos resultados se presentan en el apartado 6 de resultados.

5.4.1 Factores

Para el presente proyecto, se realizaron 6 diseños experimentales factoriales. Dos para el caso determinístico (uno para tardanza y otro para *flowtime*) y otros cuatro para el caso estocástico (uno por función objetivo). En la Tabla 4 se presentan los factores analizados y sus niveles correspondientes para el caso determinístico y en la Tabla 5 para el caso estocástico.

Tabla 4: Factores y niveles del diseño experimental para el caso determinístico

| Factor | | Descripción | Notación | Niveles | | |
|---------------------|-------------|---|----------|---------|-----|-----|
| Factores de interés | Alfa | Para la ejecución del GRASP se tomaron tres diferentes alfas (0.2, 0.5 y 0.7), los cuales fueron las tres mejores alfas obtenidos a partir de una prueba inicial (ver apartado 6). | α | 0.2 | 0.5 | 0.7 |
| | Iteraciones | La metaheurística se corrió con tres cantidades de iteraciones (100, 200 y 300), las cuales se determinaron a partir de una prueba inicial (ver apartado 6). | I | 100 | 200 | 300 |
| Factor de Bloqueo | Instancias | Se tomaron 324 instancias, con ambientes específicos de n trabajos, m máquinas y e estaciones; 108 instancias de 10, 15 y 20 trabajos respectivamente, donde las instancias con 2 estaciones tenían 4 o 6 máquinas, las de 3 estaciones 6 o 9 máquinas y las de 4 estaciones 8 o 12 máquinas. | Inst | 1 a 324 | | |

Tabla 5: Factores y niveles del diseño experimental para el caso estocástico

| Factor | | Descripción | Notación | Niveles | | |
|---------------------|---|---|----------|-----------|--------------|--|
| Factor de Bloqueo | Instancias | Se tomaron 324 instancias, con ambientes específicos de N trabajos, M máquinas y E estaciones; 108 instancias de 10, 15 y 20 trabajos respectivamente, donde las instancias con 2 estaciones tenían 4 o 6 máquinas, las de 3 estaciones 6 o 9 máquinas y las de 4 estaciones 8 o 12 máquinas. | Inst | 1 a 324 | | |
| Factores de interés | Valor esperado para MTTR | Para obtener el valor esperado del MTTR se tomaron dos niveles \bar{P} y $0,1\bar{P}$. | P | \bar{P} | $0,1\bar{P}$ | |
| | Porcentaje de tiempo que las máquinas presentan daños | Porcentaje de tiempo en donde las máquinas pueden llegar a presentar daños y afectar la producción. | Ag | 5% | 15% | |
| | Distribución probabilidad MTTR | Distribución de probabilidad utilizada para generar las realizaciones de los tiempos de duración de los daños en las máquinas | D_MTTR | Lognormal | Uniforme | |
| | Distribución de probabilidad del MTBF | Distribución de probabilidad utilizada para generar las realizaciones de los tiempos entre daños de las máquinas | D_MTBF | Lognormal | Uniforme | |

| Factor | Descripción | Notación | Niveles | |
|--------------------------------|--|----------|---------|-----|
| Coefficiente de variación MTTR | Coefficiente de variación utilizado para generar las realizaciones de los tiempos de duración de los daños en las máquinas | CV_MTTR | 0.2 | 0.4 |
| Coefficiente de variación MTBF | Coefficiente de variación utilizado para generar las realizaciones de los tiempos entre daños de las máquinas | CV_MTBF | 0.2 | 0.4 |

Para el diseño experimental del caso determinístico se obtuvieron un total de 2916 observaciones. La variable de respuesta utilizada es el porcentaje de mejoramiento de la metaheurística con respecto a los resultados obtenidos por las reglas de despacho ENS2 y FL (4).

$$\text{porcentaje de mejoramiento} = \frac{\text{Regla de despacho} - \text{Función objetivo GRASP}}{\text{Regla de despacho}} * 100 \quad (4)$$

Para el caso estocástico se realizaron 4 ANOVAS una para cada función objetivo. Por cada ANOVA se obtuvieron un total de 20736 observaciones.

5.4.2 Variables de respuesta

Las hipótesis planteadas para cada uno de los factores y sus respectivas interacciones en el apartado anterior se evaluaron para todas las instancias, donde las variables de respuesta corresponden en el caso determinístico a los % de mejoramiento de la tardanza y el *flowtime* con respecto a las reglas de despacho, mientras que en el caso estocástico corresponden al valor esperado y desviación estándar de la tardanza y al valor esperado y desviación estándar del *flowtime*.

5.5 Restricciones

El desarrollo de la investigación se presenta en un ambiente FFS con las siguientes restricciones: los tiempos MTTR y MTBF se modelarán a través de dos distribuciones de probabilidad uniforme y lognormal, con dos coeficientes de variación (0.2 y 0.4). En este modelo se trabaja sin *preemption*, la recirculación no es permitida, y todos los trabajos se encontrarán disponibles desde el inicio de la programación para ser procesados. El problema no tiene en cuenta los tiempos de alistamiento. Los tiempos de procesamiento y fechas de entrega son conocidos y determinísticos. Adicionalmente, el modelo de *scheduling* utilizado para este caso de estudio es el *nonresumable*, el cual de acuerdo con Kubzin & Strusevich (2004) y Rajkumar, Asokan, & Vamsikrishna (2010) recibe este nombre, puesto que, al dañarse una máquina mientras un trabajo está siendo procesado, dicha operación se ve afectada y debe reiniciarse tan pronto la máquina vuelve a estar disponible.

5.6 Cumplimiento del estándar

Se realizó un diseño que cumpliera con cada uno de los siguientes estándares, tal como se presenta en la Tabla 6.

Tabla 6: Estándar de diseño

| Estándar | Descripción | Cumplimiento |
|----------------------|--|--|
| Números aleatorios | A partir de la norma ISO 28640, los números aleatorios son considerados variables aleatorias. | Los números aleatorios utilizados en la metaheurística y simheurística se realizaron a través de la función: Rnd y estos dependen del generador de VBA, el cual cumple con la norma. |
| Metaheurística GRASP | El diseño de la metaheurística consta en llevar a cabo dos etapas, con el fin de obtener dos soluciones diferentes en cada | Para cada etapa, se adaptó la metaheurística propuesta por los autores Feo & Resende (1995) al problema planteado de la siguiente manera: |

| | | |
|--|--|--|
| | <p>una de ellas y seleccionar aquella con mejor desempeño en cuanto al valor de cada función objetivo, estas son:</p> <ol style="list-style-type: none"> 1. Fase de construcción (teniendo como funciones <i>greedy</i> las reglas de despacho EDD y SPT) 2. Fase de búsqueda local | <ol style="list-style-type: none"> 1. Fase de construcción: Se encuentra una solución inicial con la regla de despacho EDD para las iteraciones impares y se encuentra una solución inicial con la regla de despacho SPT para las iteraciones pares. Cada uno de los trabajos a secuenciar está sujeto al vector RCL. 2. Fase de búsqueda local: Se realizan permutaciones a partir de la secuencia inicial, intercambiando la posición <i>i</i>-ésima con la posición <i>j</i>-ésima de la misma. |
| Simheurística | <p>Uno de los principales desafíos que tiene la simulación híbrida es dar respuesta a la incertidumbre. Este aspecto es abordado por una variedad de enfoques y entre ellos la programación estocástica (Figueira & Almada-Lobo, 2014). De acuerdo a los autores (Juan et al., 2015) una simheurística es una metodología que integra una metaheurística con una simulación, para resolver problemas con parámetros estocásticos, la cual se basa en arrojar información de salida a partir de la aplicación de un procedimiento de optimización con un modelo de simulación, con el fin de que éste evalúe las soluciones obtenidas a través de este proceso.</p> | <p>De acuerdo con lo planteado por los autores que han estudiado el uso de una simheurística, se integró la metaheurística GRASP con una simulación de Monte Carlo, con el fin de obtener soluciones robustas para el problema FFS. En primera instancia, al realizar la simulación se generan los daños en las máquinas: MTTR y MTBF. Posteriormente, se calculan las cuatro funciones objetivo teniendo en cuenta estos daños por medio del proceso de optimización de la metaheurística.</p> |
| Construcción frontera de Pareto | <p>Se construye una frontera de Pareto del valor esperado de la tardanza, la desviación estándar de la tardanza, el valor esperado del <i>flowtime</i> y la desviación estándar del <i>flowtime</i> en el ambiente FFS.</p> | <p>Se garantiza la construcción de la frontera de Pareto teniendo en cuenta la metodología (1+1)-PAES propuesta por los autores Knowles & Corne (2000), en donde se genera un archivo que guarda las soluciones encontradas que no están dominadas (NDS por sus siglas en inglés) para cada función objetivo.</p> |
| Robustez de las soluciones | <p>Según (Feng et al., 2016) una solución robusta a menudo se ve afectada por aquellos encargados en tomar decisiones. Dada esta situación, la robustez se encuentra expuesta al riesgo y es por esta razón que se centra en cubrirse contra el peor rendimiento. Esto se logra a partir de la minimización de la desviación estándar de cierta función objetivo cuando se implementa una simulación (Li & Ierapetritou, 2008).</p> | <p>Las soluciones robustas se obtienen a partir de dos procedimientos: al aplicar la simulación de Monte Carlo y al ser evaluadas por la metodología PAES, es decir, cuando se evalúan la desviación estándar y valor esperado tanto de la tardanza como del <i>flowtime</i>, en donde se tiene como criterio la dominancia para determinar si dicha solución puede entrar o no al archivo de Pareto.</p> |
| Número de iteraciones del algoritmo determinístico | <p>El número de iteraciones permite tener una cantidad considerable de repeticiones de la metaheurística, con el fin de ver su evolución en cada repetición y, a su vez, que ésta pueda ser comparada con la simheurística.</p> | <p>Se realizó una prueba inicial con cinco iteraciones, variando el alfa desde 0,1 a 1 para determinar el número de iteraciones a realizar. El cumplimiento de este estándar puede ser observado en el apartado 6 de resultados.</p> |

| | | |
|---|---|---|
| Número de iteraciones de la simheurística | La simheurística busca demostrar que en un entorno real siempre se está expuesto a un cierto nivel de aleatoriedad, por lo que se hace necesario realizar un número de iteraciones que encuentre soluciones que reflejen un comportamiento probabilístico similar al que se puede encontrar en una industria. | Se determina el número de iteraciones para la simheurística según los resultados del diseño experimental para el caso determinístico. El cumplimiento de este estándar puede ser observado en el apartado 6 de resultados. |
|---|---|---|

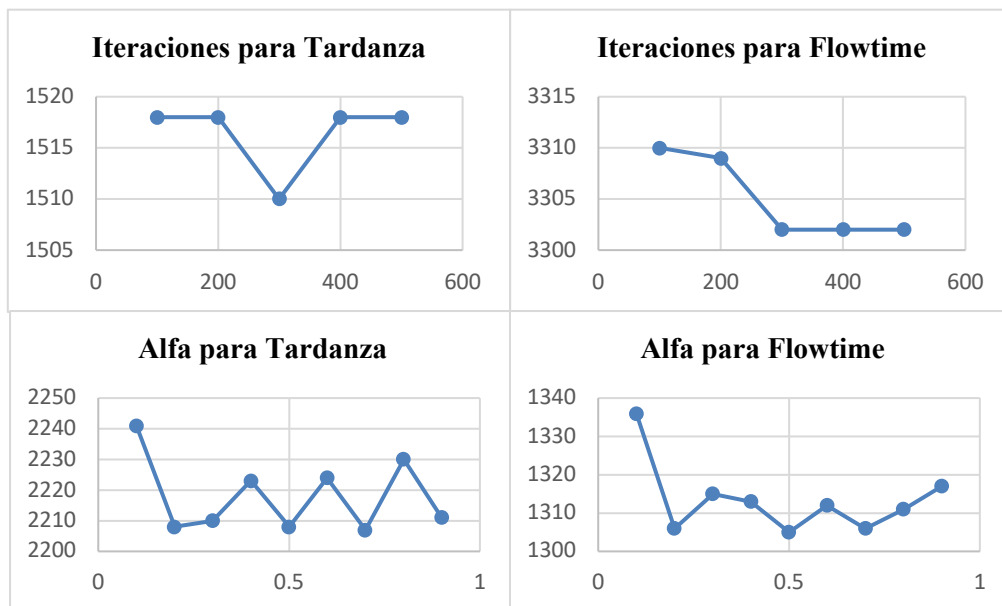
6. Resultados

A continuación, se presentan los resultados obtenidos a partir de las pruebas realizadas a cada caso. Para la ejecución de dichas pruebas, se utilizó el *software* estadístico SPSS. En el **Anexo 1** se pueden encontrar las pruebas realizadas con sus respectivos resultados del caso determinístico y para el caso estocástico en el **Anexo 2**.

6.1 Desempeño de la metodología propuesta en el caso determinístico

Inicialmente, para establecer el valor del parámetro (α) del GRASP y la cantidad de iteraciones, se ejecutó el algoritmo determinístico para 5 instancias aleatorias, con distintos valores: α entre 0 y 1 (en incrementos de 0.1) e iteraciones de 100 a 500 (en incrementos de 100). De esta prueba inicial, se tomaron los 3 mejores resultados para proceder a correr todas las instancias. Los valores escogidos para α fueron: 0.2, 0.5 y 0.7, y para la cantidad de iteraciones: 100, 200 y 300. A continuación, en la Ilustración 3 se muestran las gráficas donde se evidencian los resultados de dichas pruebas.

Ilustración 2: Resultados obtenidos para alfa e iteraciones



Para la realización del diseño experimental, se comprueban los supuestos de normalidad y homogeneidad de varianzas de los datos obtenidos para cada variable, usando un nivel de significancia del 5%. Tanto la prueba de Kolomogorov – Smirnov como la prueba de Levene, indican el incumplimiento del supuesto de normalidad y homocedasticidad respectivamente, puesto que para ambos casos se obtiene un valor – p para cada variable menor al 5%. Los resultados de dichas pruebas se pueden observar en el **Anexo 1**.

A pesar de que no se cumplan con los supuestos de normalidad y homocedasticidad, se realizó un análisis de varianza para evaluar los efectos de los factores anteriormente mencionados y sus interacciones para cada una de las funciones objetivo.

6.1.1 Resultados del diseño experimental para la tardanza y *flowtime*

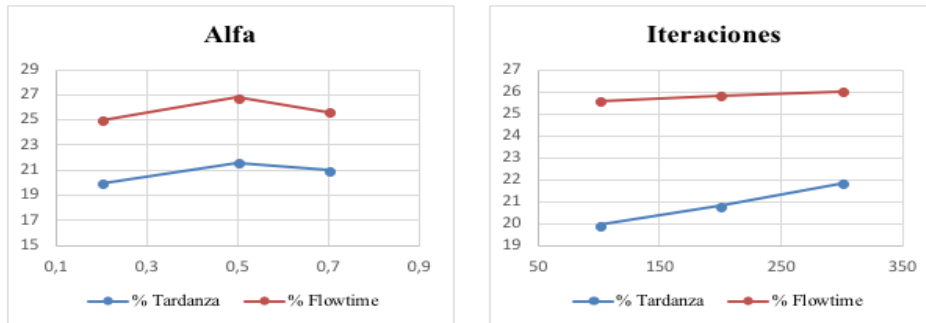
En este apartado, se realizan pruebas de normalidad teniendo en cuenta un número total de datos de 2916. El análisis de varianza para la tardanza y *flowtime* se puede observar en la Tabla 7.

Tabla 7: Resultados del ANOVA para el porcentaje de mejoramiento de la tardanza con respecto al ENS2 y el porcentaje de mejoramiento del *flowtime* con respecto al FL

| Factor | Valor – p Tardanza | Decisión | Valor – p <i>flowtime</i> | Decisión |
|-------------------------|----------------------------------|--------------------------------|---|---------------|
| Alfa | 0.036 | Se rechaza | 0.003 | Se rechaza |
| It | 0.011 | Se rechaza | 0.711 | No se rechaza |
| Bloque (Inst) | 0.000 | Se rechaza | 0.000 | Se rechaza |
| Alfa - It | 0.542 | No se rechaza | 0.998 | No se rechaza |
| Resumen del modelo | | | | |
| R ² Tardanza | R ² ajustado Tardanza | R ² <i>flowtime</i> | R ² ajustado <i>flowtime</i> | |
| 99.99% | 99.99% | 94.1% | 93.3% | |

De acuerdo con los resultados del ANOVA, para el porcentaje de mejoramiento de la tardanza, todos los factores son significativos, mientras que para el porcentaje de mejoramiento del *flowtime*, el factor iteración no tiene efecto significativo. Por otro lado, con una prueba de diferencia de medias para las dos funciones objetivo, se determinó que el alfa debe ser 0.5 y que 300 sería la cantidad de iteraciones para ejecutar el algoritmo GRASP para el caso estocástico. En la Ilustración 4 se presenta el comportamiento del porcentaje de mejoramiento de los factores alfa e iteraciones.

Ilustración 4: Pruebas de perfil para alfa e iteraciones



Debido al incumplimiento de supuestos en la prueba de ANOVA, se realiza la prueba no paramétrica de Friedman, la cual determina que factores tienen efecto significativo sobre las funciones objetivo (Tabla 8).

Tabla 8: Prueba de Friedman para el porcentaje de mejoramiento de la tardanza con respecto al ENS2 y del *flowtime* con respecto al FL

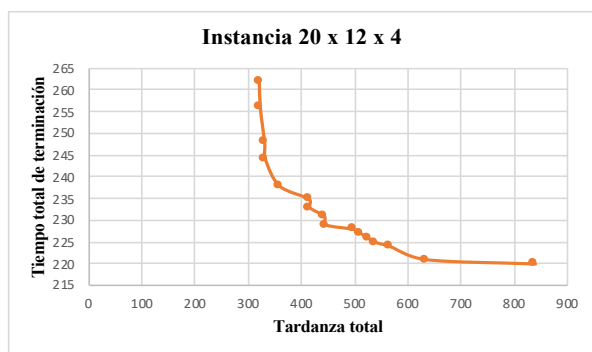
| Prueba de Friedman | | | | | |
|--------------------|-----|----------------|----|--------------------|---------------------|
| Factor | N | X ² | gl | Valor p % mejora T | Valor p % mejora FT |
| Alfa | 972 | 22.11 | 2 | 0.000 | 0.000 |
| Iteración | 972 | 56.6 | 2 | 0.000 | 0.000 |

Los valores-p resultantes que son menos a la significancia del 5% indican que las hipótesis nulas se rechazan, es decir, que tanto los factores alfa como número de iteraciones tienen un efecto significativo en el porcentaje de mejoramiento de la tardanza con respecto al ENS2 y del *flowtime* con respecto a la regla de despacho FL.

6.1.2 Frontera de Pareto

Teniendo en cuenta que uno de los objetivos de la metaheurística desarrollada es obtener la frontera de Pareto para cada instancia, se presenta a continuación un ejemplo gráfico para una instancia con 20 trabajos, 12 máquinas y 4 estaciones, donde se observa un comportamiento inverso de los dos objetivos evaluados (ver Ilustración 5).

Ilustración 5: Frontera de Pareto para las funciones objetivo tardanza y flowtime



6.2 Desempeño de la metodología propuesta en el caso estocástico

Tal como se realizó en el caso determinístico, se comprueban los supuestos de normalidad y homogeneidad de varianzas de los datos obtenidos para cada variable usando un nivel de significancia de 5%. La prueba de Kolomogorov – Smirnov y la prueba de Levene indican el incumplimiento del supuesto de normalidad y homocedasticidad respectivamente, puesto que para ambos casos se obtiene un valor – p para cada variable menor al 5%. Los resultados de dichas pruebas se pueden observar en el **Anexo 2**.

Inicialmente, se evaluaron los efectos de los factores para la simheurística junto con sus interacciones para cada uno los objetivos por medio de un ANOVA.

6.2.1 Resultados del diseño experimental para el valor esperado de la tardanza, valor esperado del flowtime, desviación estándar de la tardanza y desviación estándar del flowtime

En este apartado, se realiza la prueba de análisis de varianza para las cuatro funciones objetivo, los resultados se pueden observar en la Tabla 9.

Tabla 9: Análisis de varianza para el valor esperado de la tardanza, valor esperado del flowtime, desviación estándar de la tardanza y desviación estándar del flowtime

| Fuente | Valor – p Valor esperado tardanza | Valor – P Valor esperado flowtime | Valor – p Desviación estándar tardanza | Valor – p Desviación estándar flowtime |
|--------------------|--|--|---|---|
| Inst (bloqueo) | 0.000 | 0.000 | 0.000 | 0.000 |
| P | 0.000 | 0.000 | 0.000 | 0.000 |
| Ag | 0.000 | 0.000 | 0.042 | 0.000 |
| CV_MTTR | 0.000 | 0.000 | 0.415 | 0.920 |
| CV_MTBFB | 0.000 | 0.000 | 0.287 | 0.419 |
| D_MTTR | 0.000 | 0.000 | 0.000 | 0.000 |
| D_MTBFB | 0.000 | 0.000 | 0.000 | 0.000 |
| Ag * CV_MTBFB | 0.374 | 0.082 | 0.642 | 0.402 |
| Ag * CV_MTTR | 0.722 | 0.579 | 0.699 | 0.811 |
| Ag * D_MTBFB | 0.000 | 0.000 | 0.869 | 0.000 |
| Ag * D MTTR | 0.026 | 0.642 | 0.515 | 0.964 |
| P * Ag | 0.000 | 0.000 | 0.000 | 0.000 |
| CV_MTBFB * CV_MTTR | 0.972 | 0.534 | 0.417 | 0.846 |
| CV_MTBFB * D_MTBFB | 0.000 | 0.000 | 0.262 | 0.421 |

| | | | | |
|--|-------|-------|-------|-------|
| CV_MTBF * D_MTTR | 0.021 | 0.000 | 0.040 | 0.673 |
| P * CV_MTBF | 0.000 | 0.000 | 0.865 | 0.249 |
| CV_MTTR * D_MTBF | 0.000 | 0.008 | 0.185 | 0.157 |
| CV_MTTR * D_MTTR | 0.000 | 0.000 | 0.451 | 0.640 |
| P * CV_MTTR | 0.000 | 0.000 | 0.589 | 0.889 |
| D_MTTR * D_MTBF | 0.005 | 0.000 | 0.000 | 0.000 |
| P * D_MTBF | 0.000 | 0.000 | 0.000 | 0.000 |
| P * D_MTTR | 0.000 | 0.000 | 0.000 | 0.000 |
| Ag * CV_MTBF * CV_MTTR | 0.965 | 0.814 | 0.770 | 0.790 |
| Ag * CV_MTBF * D_MTBF | 0.433 | 0.066 | 0.560 | 0.336 |
| Ag * CV_MTBF * D_MTTR | 0.468 | 0.916 | 0.962 | 0.407 |
| P * Ag * CV_MTBF | 0.251 | 0.867 | 0.828 | 0.659 |
| Ag * CV_MTTR * D_MTBF | 0.697 | 0.185 | 0.616 | 0.490 |
| Ag * CV_MTTR * D_MTTR | 0.843 | 0.851 | 0.895 | 0.519 |
| P * Ag * CV_MTTR | 0.872 | 0.520 | 0.680 | 0.776 |
| Ag * D_MTTR * D_MTBF | 0.000 | 0.000 | 0.129 | 0.836 |
| P * Ag * D_MTBF | 0.000 | 0.000 | 0.000 | 0.000 |
| P * Ag * D_MTTR | 0.368 | 0.291 | 0.860 | 0.664 |
| CV_MTBF * CV_MTTR * D_MTBF | 0.976 | 0.646 | 0.455 | 0.764 |
| CV_MTBF * CV_MTTR * D_MTTR | 0.825 | 0.436 | 0.476 | 0.737 |
| P * CV_MTBF * CV_MTTR | 0.951 | 0.482 | 0.413 | 0.914 |
| CV_MTBF * D_MTTR * D_MTBF | 0.020 | 0.000 | 0.029 | 0.670 |
| P * CV_MTBF * D_MTBF | 0.000 | 0.000 | 0.871 | 0.247 |
| P * CV_MTBF * D_MTTR | 0.058 | 0.000 | 0.047 | 0.599 |
| CV_MTTR * D_MTTR * D_MTBF | 0.000 | 0.006 | 0.228 | 0.199 |
| P * CV_MTTR * D_MTBF | 0.000 | 0.001 | 0.337 | 0.200 |
| P * CV_MTTR * D_MTTR | 0.000 | 0.000 | 0.644 | 0.956 |
| P * D_MTTR * D_MTBF | 0.000 | 0.000 | 0.000 | 0.000 |
| Ag * CV_MTBF * CV_MTTR * D_MTBF | 0.986 | 0.616 | 0.770 | 0.988 |
| Ag * CV_MTBF * CV_MTTR * D_MTTR | 0.993 | 0.679 | 0.719 | 0.843 |
| P * Ag * CV_MTBF * CV_MTTR | 0.998 | 0.665 | 0.633 | 0.887 |
| Ag * CV_MTBF * D_MTTR * D_MTBF | 0.509 | 0.675 | 0.911 | 0.514 |
| P * Ag * CV_MTBF * D_MTBF | 0.340 | 0.883 | 0.847 | 0.629 |
| P * Ag * CV_MTBF * D_MTTR | 0.522 | 0.930 | 0.629 | 0.599 |
| Ag * CV_MTTR * D_MTTR * D_MTBF | 0.876 | 0.234 | 0.564 | 0.448 |
| P * Ag * CV_MTTR * D_MTBF | 0.686 | 0.220 | 0.647 | 0.598 |
| P * Ag * CV_MTTR * D_MTTR | 0.953 | 0.629 | 0.725 | 0.685 |
| P * Ag * D_MTTR * D_MTBF | 0.000 | 0.000 | 0.440 | 0.502 |
| CV_MTBF * CV_MTTR * D_MTTR * D_MTBF | 0.916 | 0.525 | 0.427 | 0.858 |
| P * CV_MTBF * CV_MTTR * D_MTBF | 0.891 | 0.344 | 0.492 | 0.678 |
| P * CV_MTBF * CV_MTTR * D_MTTR | 0.939 | 0.461 | 0.496 | 0.901 |
| P * CV_MTBF * D_MTTR * D_MTBF | 0.504 | 0.000 | 0.075 | 0.627 |
| P * CV_MTTR * D_MTTR * D_MTBF | 0.000 | 0.003 | 0.244 | 0.168 |
| Ag * CV_MTBF * CV_MTTR * D_MTTR * D_MTBF | 0.980 | 0.611 | 0.691 | 0.920 |
| P * Ag * CV_MTBF * CV_MTTR * D_MTBF | 0.948 | 0.818 | 0.672 | 0.923 |
| P * Ag * CV_MTBF * CV_MTTR * D_MTTR | 0.970 | 0.781 | 0.866 | 0.861 |
| P * Ag * CV_MTBF * D_MTTR * D_MTBF | 0.428 | 0.960 | 0.625 | 0.753 |
| P * Ag * CV_MTTR * D_MTTR * D_MTBF | 0.731 | 0.262 | 0.647 | 0.658 |
| P * CV_MTBF * CV_MTTR * D_MTTR * D_MTBF | 0.973 | 0.404 | 0.485 | 0.793 |
| P * Ag * CV_MTTR * CV_MTBF * D_MTTR * D_MTBF | 0.999 | 0.864 | 0.808 | 0.874 |

Resumen del modelo

| Función objetivo | R ² | R ² ajustado |
|---|----------------|-------------------------|
| Valor esperado de la tardanza | 86.8% | 86.5% |
| Valor esperado del <i>flowtime</i> | 90.8% | 90.7% |
| Desviación estándar de la tardanza | 26.7% | 25.3% |
| Desviación estándar del <i>flowtime</i> | 25.3% | 23.9% |

A partir de los resultados obtenidos con el ANOVA, los coeficientes de variación tanto del MTTR como del MTBF no son factores significativos para las desviaciones estándar de tardanza y *flowtime*. En cambio, para los valores esperados de tardanza y *flowtime*, los 7 factores evaluados individualmente tienen efecto significativo.

Debido a que no se cumplen los supuestos de normalidad y homocedasticidad, se aplica la prueba no paramétrica de Friedman para determinar qué factores inciden en las variables de respuesta. A continuación, en la Tabla 10 se muestra la prueba de Friedman realizada para las cuatro funciones objetivo en donde los valores p que se resaltan son aquellos que están por debajo del 5%, es decir, que tienen efecto significativo sobre las variables de respuesta del experimento.

Tabla 10: Prueba de Friedman para el valor esperado de la tardanza, valor esperado del flowtime, desviación estándar de la tardanza y desviación estándar del flowtime

| Prueba de Friedman | | | | | | | | | | |
|--------------------|-------|----|----------|----------|---------|---------|---------|-------|-------|-------|
| Factor | N | gl | X^2 | | | | Valor p | | | |
| | | | VE T | VE FT | DE T | DE FT | VE T | VE FT | DE T | DE FT |
| P | 10368 | 1 | 0.000 | 0.105 | 532.950 | 830.401 | 0.984 | 0.746 | 0.000 | 0.000 |
| Ag | 10368 | 1 | 9043.972 | 7439.259 | 52.861 | 37.932 | 0.000 | 0.000 | 0.000 | 0.000 |
| CV MTTR | 10368 | 1 | 83.710 | 29.189 | 13.220 | 4.537 | 0.000 | 0.000 | 0.000 | 0.033 |
| CV MTBF | 10368 | 1 | 469.471 | 123.581 | 0.041 | 0.716 | 0.000 | 0.000 | 0.840 | 0.398 |
| D MTTR | 10368 | 1 | 240.967 | 120.605 | 0.013 | 45.148 | 0.000 | 0.000 | 0.909 | 0.000 |
| D MTBF | 10368 | 1 | 2548.187 | 2277.407 | 126.865 | 608.883 | 0.000 | 0.000 | 0.000 | 0.000 |

A partir de la tabla anterior, se infiere que, en términos generales, los 6 parámetros de interés tienen un efecto significativo sobre las funciones objetivo, debido al rechazo de la hipótesis nula en donde el valor-p es menor al 5%. Particularmente, se tiene que los factores: Ag, coeficiente de variación del MTTR y la distribución de probabilidad del MTBF tienen efecto significativo en las cuatro funciones objetivo. Mientras que, el valor esperado del MTTR (P) es significativo sólo para las desviaciones estándar, el coeficiente de variación del MTBF es significativo solo para los valores esperados y la distribución de probabilidad del MTTR no es significativo para la desviación estándar de la tardanza.

6.2.1 Análisis descriptivo

Se calcularon estadísticos descriptivos de tendencia central, dispersión, distribución y según los valores percentiles, para cada nivel de cada factor de interés seleccionado para el caso estocástico, los cuales se pueden encontrar en el **Anexo 3**. A continuación, en la Tabla 11 y en la Tabla 12, se presenta la media y la desviación estándar por tipo de instancias, es decir, las posibles combinaciones entre trabajos (10, 15 y 20) y estaciones (2, 3 y 4).

Tabla 11: Estadístico descriptivo - media

| Trabajos Factor | | MEDIA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|-------|--------|---------|---------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|------|------|------|------|------|-------|-------|
| | | VE_T | | | | | | | | | VE_FT | | | | | | | | | DE_T | | | | | | | | | DE_FT | | | | | | | | |
| | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | |
| | | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 |
| P | 0.1 | 704.86 | 3935.08 | 2301.02 | 1400.58 | 1555.01 | 2605.23 | 1054.70 | 1716.06 | 2643.48 | 312.63 | 441.40 | 536.16 | 359.27 | 467.37 | 563.52 | 411.46 | 503.04 | 607.91 | 9.58 | 31.58 | 17.58 | 6.72 | 7.96 | 27.26 | 5.93 | 13.67 | 35.70 | 2.98 | 3.86 | 1.97 | 1.58 | 1.11 | 3.10 | 1.82 | 2.02 | 3.12 |
| | 1 | 725.66 | 3589.34 | 2008.30 | 1302.33 | 1340.78 | 2263.02 | 982.52 | 1566.97 | 2521.23 | 311.17 | 404.55 | 487.89 | 355.92 | 430.79 | 514.92 | 410.60 | 486.29 | 587.95 | 47.33 | 70.61 | 54.96 | 18.33 | 14.20 | 42.47 | 12.67 | 24.06 | 111.31 | 16.23 | 12.43 | 8.73 | 5.68 | 3.24 | 6.62 | 4.98 | 24.06 | 17.87 |
| Ag | 5% | 520.44 | 3038.35 | 1526.94 | 1138.09 | 1025.60 | 1696.21 | 827.16 | 1214.25 | 1816.03 | 283.79 | 377.65 | 461.95 | 333.11 | 410.10 | 486.95 | 384.69 | 455.87 | 544.37 | 31.19 | 22.84 | 31.47 | 11.72 | 8.70 | 32.54 | 9.85 | 19.02 | 68.93 | 9.64 | 1.45 | 3.70 | 2.83 | 1.32 | 4.66 | 3.08 | 3.06 | 9.20 |
| | 15% | 849.83 | 4265.92 | 2782.38 | 1564.81 | 1870.19 | 3172.04 | 1210.06 | 2068.78 | 3348.67 | 332.15 | 459.39 | 562.10 | 382.09 | 488.06 | 591.49 | 437.38 | 533.46 | 651.49 | 27.03 | 50.81 | 41.07 | 13.33 | 13.46 | 37.19 | 8.74 | 18.72 | 78.08 | 10.35 | 8.92 | 6.99 | 4.43 | 3.03 | 5.05 | 3.72 | 3.87 | 11.79 |
| CV MTTR | 0.2 | 695.67 | 3787.08 | 2182.88 | 1364.53 | 1469.39 | 2457.85 | 1027.40 | 1716.06 | 2607.23 | 309.38 | 424.44 | 513.87 | 359.37 | 450.93 | 540.46 | 411.81 | 503.04 | 597.91 | 30.56 | 50.62 | 35.96 | 13.78 | 10.78 | 31.48 | 9.98 | 13.67 | 69.57 | 10.37 | 7.97 | 5.25 | 3.90 | 2.24 | 4.44 | 3.80 | 2.02 | 9.81 |
| | 0.4 | 674.61 | 3737.33 | 2126.45 | 1338.38 | 1426.40 | 2410.41 | 1009.82 | 1619.59 | 2557.48 | 306.56 | 421.50 | 510.18 | 355.83 | 447.23 | 537.98 | 410.26 | 492.80 | 597.95 | 27.66 | 51.57 | 36.58 | 11.26 | 11.38 | 38.25 | 8.62 | 18.61 | 77.44 | 9.61 | 8.31 | 5.45 | 3.37 | 2.12 | 5.28 | 3.00 | 3.29 | 11.18 |
| CV MTBF | 0.2 | 664.69 | 3701.53 | 2084.74 | 1332.16 | 1403.47 | 2370.73 | 994.72 | 1587.88 | 2491.13 | 305.75 | 419.42 | 509.58 | 356.47 | 446.31 | 536.99 | 408.30 | 491.26 | 593.69 | 31.09 | 53.08 | 34.48 | 15.30 | 10.04 | 31.27 | 9.05 | 18.38 | 67.67 | 10.70 | 8.39 | 5.21 | 4.47 | 2.07 | 4.46 | 3.54 | 3.50 | 10.01 |
| | 0.4 | 705.58 | 3822.89 | 2224.58 | 1370.75 | 1492.32 | 2497.53 | 1042.50 | 1695.15 | 2673.58 | 310.19 | 426.53 | 514.47 | 358.73 | 451.85 | 541.45 | 413.76 | 498.06 | 602.17 | 27.12 | 49.11 | 38.06 | 9.74 | 12.12 | 38.46 | 9.55 | 19.36 | 79.33 | 9.29 | 7.90 | 5.49 | 2.79 | 2.28 | 5.26 | 3.26 | 3.42 | 10.99 |
| D MTTR | Unif. | 699.47 | 3837.69 | 2245.73 | 1362.88 | 1514.05 | 2485.54 | 996.43 | 1631.41 | 2510.63 | 302.07 | 421.00 | 512.22 | 347.98 | 448.04 | 532.84 | 395.28 | 481.02 | 575.79 | 22.46 | 44.63 | 33.41 | 12.18 | 14.06 | 18.46 | 7.89 | 11.32 | 25.82 | 5.74 | 5.48 | 4.35 | 3.33 | 2.93 | 2.84 | 3.07 | 2.49 | 3.55 |
| | LogN. | 670.80 | 3686.72 | 2063.60 | 1340.02 | 1381.74 | 2382.72 | 1040.79 | 1651.62 | 2654.08 | 313.87 | 424.95 | 511.83 | 367.22 | 450.12 | 545.60 | 426.78 | 508.31 | 620.07 | 35.75 | 57.56 | 39.13 | 12.87 | 8.10 | 51.27 | 10.70 | 26.41 | 121.19 | 14.25 | 10.80 | 6.35 | 3.93 | 1.42 | 6.87 | 3.72 | 4.43 | 17.44 |
| D MTBF | Unif. | 454.34 | 3171.38 | 1611.09 | 1068.48 | 1046.87 | 1803.29 | 771.65 | 1151.43 | 1740.12 | 250.35 | 350.84 | 427.42 | 306.54 | 377.22 | 444.61 | 363.91 | 416.13 | 483.30 | 14.55 | 46.08 | 21.87 | 15.73 | 10.56 | 17.75 | 11.89 | 10.92 | 12.66 | 6.88 | 8.08 | 6.20 | 4.73 | 3.22 | 4.10 | 5.03 | 3.69 | 4.65 |
| | LogN. | 915.93 | 4353.04 | 2698.24 | 1634.42 | 1848.92 | 3064.97 | 1265.57 | 2131.60 | 3426.24 | 365.59 | 495.11 | 596.63 | 408.66 | 520.94 | 633.83 | 458.16 | 573.19 | 712.85 | 43.67 | 56.12 | 50.67 | 9.31 | 11.60 | 51.99 | 6.71 | 26.82 | 134.93 | 13.10 | 8.20 | 4.50 | 2.53 | 1.14 | 5.62 | 1.77 | 3.23 | 16.47 |

Tabla 12: Estadístico descriptivo – desviación estándar

| Trabajos Factor | | DESVIACIÓN ESTÁNDAR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|-------|---------------------|---------|---------|---------|---------|---------|--------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|--------|--------|-------|-------|--------|-------|-------|--------|-------|-------|-------|------|------|-------|------|------|-------|
| | | VE_T | | | | | | | | | VE_FT | | | | | | | | | DE_T | | | | | | | | | DE_FT | | | | | | | | |
| | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | | 2 | | | 3 | | | 4 | | |
| | | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 | 10 | 15 | 20 |
| P | 0.1 | 4.69 | 1429.38 | 2046.65 | 954.86 | 1176.62 | 2329.96 | 696.44 | 1175.29 | 2184.25 | 67.27 | 110.53 | 121.63 | 76.21 | 95.28 | 121.24 | 64.46 | 87.86 | 119.67 | 13.49 | 48.31 | 35.64 | 12.03 | 12.52 | 39.39 | 10.69 | 18.74 | 50.65 | 4.60 | 5.54 | 3.59 | 3.10 | 1.54 | 3.92 | 3.77 | 3.15 | 4.69 |
| | 1 | 694.08 | 1666.76 | 2079.07 | 1037.92 | 1261.54 | 2373.65 | 811.65 | 1379.74 | 2550.51 | 133.80 | 175.01 | 193.13 | 130.52 | 161.61 | 217.44 | 130.34 | 173.02 | 266.96 | 68.57 | 148.11 | 110.78 | 30.87 | 20.20 | 156.52 | 22.77 | 50.73 | 349.18 | 22.64 | 27.56 | 14.19 | 9.64 | 4.69 | 25.09 | 8.97 | 8.08 | 59.53 |
| Ag | 5% | 465.20 | 1255.49 | 1711.44 | 901.19 | 963.91 | 1910.66 | 669.04 | 1067.19 | 1934.74 | 98.66 | 137.49 | 148.16 | 102.66 | 125.59 | 163.93 | 96.63 | 131.22 | 198.50 | 57.07 | 110.12 | 77.71 | 29.20 | 12.57 | 100.80 | 18.32 | 37.67 | 250.41 | 18.67 | 18.92 | 8.72 | 8.23 | 2.01 | 15.70 | 6.24 | 5.35 | 39.41 |
| | 15% | 553.85 | 1672.39 | 2199.18 | 1044.32 | 1308.74 | 2526.05 | 790.84 | 1338.57 | 2522.84 | 102.00 | 148.20 | 162.13 | 105.40 | 130.57 | 175.55 | 102.06 | 132.53 | 201.61 | 50.40 | 113.60 | 90.31 | 17.66 | 20.36 | 126.46 | 17.87 | 39.49 | 254.17 | 17.34 | 21.62 | 12.47 | 6.47 | 4.60 | 20.11 | 7.78 | 7.10 | 46.02 |
| CV MTTR | 0.2 | 547.57 | 1582.62 | 2083.54 | 1008.68 | 1238.58 | 2368.44 | 764.51 | 1175.29 | 2398.15 | 104.86 | 148.61 | 164.79 | 109.51 | 135.76 | 177.28 | 104.26 | 87.86 | 204.40 | 57.43 | 106.25 | 85.37 | 26.86 | 14.77 | 99.23 | 19.64 | 18.74 | 231.52 | 18.68 | 18.98 | 10.24 | 8.15 | 3.77 | 15.42 | 7.77 | 3.15 | 37.59 |
| | 0.4 | 526.70 | 1541.13 | 2052.18 | 987.97 | 1209.90 | 2347.53 | 749.53 | 1261.46 | 2351.84 | 101.53 | 146.40 | 161.53 | 104.16 | 132.01 | 178.12 | 101.35 | 135.35 | 209.78 | 50.03 | 117.23 | 83.38 | 20.99 | 19.13 | 127.66 | 16.40 | 38.57 | 271.51 | 17.32 | 21.60 | 11.49 | 6.66 | 3.52 | 20.32 | 6.25 | 5.99 | 47.55 |
| CV MTBF | 0.2 | 515.84 | 1514.06 | 2006.52 | 980.93 | 1185.74 | 2305.11 | 743.03 | 1240.93 | 2283.77 | 99.53 | 143.64 | 158.95 | 104.97 | 129.98 | 174.34 | 100.10 | 134.51 | 199.06 | 56.90 | 106.81 | 76.16 | 30.93 | 13.80 | 93.46 | 17.32 | 38.64 | 220.92 | 19.52 | 19.43 | 10.56 | 9.17 | 3.52 | 14.96 | 7.43 | 6.56 | 40.34 |
| | 0.4 | 557.26 | 1606.64 | 2125.62 | 1015.34 | 1260.53 | 2408.31 | 770.18 | 1323.04 | 2459.92 | 106.73 | 151.21 | 167.27 | 108.76 | 137.67 | 180.98 | 105.39 | 140.28 | 214.78 | 50.59 | 116.69 | 91.83 | 13.90 | 19.78 | 131.93 | 18.86 | 38.54 | 280.14 | 16.35 | 21.20 | 11.20 | 5.05 | 3.77 | 20.66 | 6.67 | 6.03 | 45.24 |
| D MTTR | Unif. | 544.34 | 1609.69 | 2142.35 | 1000.51 | 1275.17 | 2387.22 | 740.97 | 1280.60 | 2291.54 | 91.91 | 139.19 | 158.16 | 90.84 | 126.44 | 161.46 | 78.94 | 118.20 | 160.94 | 27.11 | 52.54 | 46.77 | 15.64 | 16.00 | 34.69 | 14.84 | 15.61 | 46.01 | 7.58 | 7.84 | 6.42 | 4.96 | 4.36 | 4.65 | 6.21 | 4.25 | 5.47 |
| | LogN. | 529.85 | 1509.48 | 1986.94 | 996.29 | 1167.93 | 2327.54 | 772.27 | 1286.84 | 2453.99 | 113.10 | 155.37 | 168.05 | 120.05 | 140.98 | 192.37 | 120.06 | 153.14 | 242.71 | 70.58 | 148.96 | 109.71 | 30.34 | 17.61 | 156.28 | 20.77 | 51.20 | 347.39 | 23.58 | 27.41 | 13.92 | 9.28 | 2.54 | 24.92 | 7.81 | 7.71 | 59.57 |
| D MTBF | Unif. | 512.98 | 1662.03 | 2105.74 | 958.49 | 1295.53 | 2399.64 | 727.06 | 1288.84 | 2289.31 | 85.98 | 136.79 | 157.37 | 92.21 | 127.41 | 166.96 | 85.70 | 121.03 | 161.89 | 16.82 | 37.47 | 30.60 | 14.35 | 11.95 | 21.07 | 15.82 | 15.14 | 20.86 | 8.32 | 8.47 | 7.79 | 6.04 | 4.51 | 4.96 | 7.56 | 6.26 | 8.08 |
| | LogN. | 455.72 | 1191.60 | 1878.45 | 956.50 | 999.28 | 2136.97 | 704.06 | 1074.34 | 2151.29 | 85.27 | 119.98 | 119.00 | 95.54 | 96.41 | 131.78 | 96.71 | 103.94 | 182.61 | 71.40 | 153.55 | 113.53 | 30.64 | 21.00 | 158.54 | 19.80 | 51.22 | 346.04 | 23.68 | 27.48 | 13.22 | 8.48 | 2.03 | 25.01 | 6.10 | 6.33 | 59.61 |

Como se puede observar en la Tabla 11, el menor resultado promedio del valor esperado de la tardanza y del *flowtime* se encuentran cuando: la distribución para el parámetro bajo incertidumbre MTBF es uniforme, el número de trabajos es 10 y el número de estaciones es 2. Mientras que, el mayor resultado promedio para ambas funciones objetivo se encuentra cuando la distribución del parámetro estocástico MTBF sigue una distribución lognormal, pero éste varía para cada función objetivo según el número de trabajos a realizar y el número de estaciones, esto es, para el caso del valor esperado de la tardanza es 15 y 2 y para el caso del valor esperado del *flowtime* es 20 y 4, respectivamente. En cambio, el menor promedio para las desviaciones estándar de la tardanza y del *flowtime*, se encuentran cuando el coeficiente del factor P toma el valor de 0.1 y de igual manera varían según el número de trabajos y estaciones. El mayor valor promedio de la desviación estándar de la tardanza y del *flowtime* se obtiene cuando la distribución del MTBF es lognormal y cuando el coeficiente del factor P toma el valor de 1, respectivamente.

Partiendo de que entre mayor sea la desviación estándar, mayor es la dispersión entre los datos, como se presenta en la Tabla 12 se obtuvo que para las cuatro funciones objetivo es menor la dispersión entre los datos con respecto a la media cuando el coeficiente del factor P es 0.1, pero teniendo en cuenta que para el valor esperado del *flowtime* y la desviación estándar de la tardanza éste mínimo se obtiene cuando se trabaja con 4 estaciones y se realizan 10 trabajos, para el valor esperado de la tardanza se obtuvo que es menor la dispersión cuando se trabaja con 2 estaciones y se realizan 10 trabajos y para la desviación estándar del *flowtime* cuando hay 3 estaciones y 15 trabajos. En cuanto a la mayor desviación típica, se encuentra en común que, para el valor esperado de la tardanza, el valor esperado y la desviación estándar del *flowtime* existe una mayor dispersión de los datos con respecto a la media cuando se deben entregar 20 trabajos, hay 4 estaciones y el coeficiente del factor P es 1.

Finalmente, de acuerdo con los resultados obtenidos se puede concluir que cada uno de los parámetros analizados afectan significativamente cada función objetivo. Esto quiere decir que, entre mayor sea el valor del parámetro, el número de trabajos y el número de estaciones, mayor será el valor de la función objetivo.

6.3 Fronteras de Pareto

Para cada instancia, en cada combinación de coeficiente de variación tanto de MTTR como de MTBF y distribución de probabilidad de MTBF y MTTR se obtuvo una frontera de Pareto. Como no es posible mostrar gráficos en cuatro dimensiones, en el **Anexo 4**, se presentan los seis gráficos de las seis posibles combinaciones de dos de los cuatro objetivos analizados, para una instancia con 20 trabajos, 6 máquinas y 3 estaciones, donde los parámetros utilizados son: $P = 0.1$, $Ag = 0.05$, $CV_MTTR = 0.2$, $CV_MTBF = 0.2$, $D_MTTR = \text{uniforme}$, $D_MTBF = \text{uniforme}$.

7. Conclusiones y recomendaciones

Este proyecto de investigación construye una propuesta encaminada a la solución de un problema de *scheduling* multi-objetivo con el fin de minimizar la desviación estándar de la tardanza, el valor esperado de la tardanza, la desviación estándar del *flowtime* y el valor esperado del *flowtime* en un ambiente FFS, con tiempos entre daños de las máquinas y la duración de los daños como parámetros estocásticos. El procedimiento para llevar esto a cabo se da a través de la hibridización de la metaheurística GRASP con la simulación de Monte Carlo y la metodología PAES, con el fin de hallar la frontera de Pareto para cada una de las funciones objetivo. Se evaluó el desempeño de la propuesta mediante la comparación de la misma con la metaheurística GRASP para el caso determinístico, teniendo en cuenta la tardanza y el *flowtime* como funciones objetivo a minimizar.

De acuerdo con el análisis realizado para el caso determinístico, los resultados arrojaron que la metaheurística propuesta presenta un porcentaje de mejoramiento en promedio del 3% en el *flowtime* y del 2% en la tardanza con respecto a las reglas de despacho FL y ENS2, respectivamente. Adicionalmente, por medio de la prueba no paramétrica de Friedman se concluye que, para las dos funciones objetivo, los factores alfa (parámetro del

GRASP), iteraciones e instancia tienen un efecto significativo sobre el porcentaje de mejoramiento de ambas funciones objetivo.

Para el caso estocástico se obtuvo por medio de la prueba no paramétrica de Friedman que los parámetros que tienen efecto significativo sobre las cuatro funciones objetivo son: el porcentaje de tiempo que las máquinas presentan daños (Ag), el coeficiente de variación del MTTR, el coeficiente de variación del MTBF y la distribución de probabilidad del MTBF. Cuando el factor (Ag) toma el valor de 5% se evidencian mejores resultados en todas las funciones objetivo, del mismo modo con el coeficiente de variación 0.2 y la distribución de probabilidad uniforme. De manera particular, el valor esperado del MTTR (P) es significativo sólo para las desviaciones estándar, cuando éste toma el valor de 0.1, se encuentran mejores resultados que cuando toma el valor de 1, debido a que la duración del daño es diez veces menor. El hecho de que las distribuciones de probabilidad y los coeficientes de variación de ambos parámetros estocásticos tengan efecto significativo en al menos dos de las cuatro funciones objetivo, muestra la importancia de hacer un ajuste correcto de las distribuciones de probabilidad de los parámetros bajo incertidumbre para obtener soluciones adecuadas.

Para investigaciones futuras, se recomienda implementar el mismo código en otro lenguaje de programación como C++, es decir, bajo programación estructurada, con la finalidad de reducir los tiempos computacionales sustancialmente. Otra alternativa, es cambiar la forma de programación por programación orientada a objetos en C++ o Java, dado que, a nivel industrial se requiere de tiempos computacionales más rápidos para la solución de este tipo de problemas.

Además, se sugiere que se utilice como criterio de parada el número de iteraciones para instancias de tamaño *small*, el cual fue implementado para este trabajo. Mientras que, para instancias de tamaño *large*, se recomienda utilizar el tiempo de ejecución como criterio de parada. Por otro lado, se propone realizar un mayor número de intercambios al implementar un tipo de búsqueda local *3-optimal* o *4-optimal* y probar la simheurística con otras distribuciones de probabilidad como la exponencial. Así mismo, se sugiere probar esta propuesta para la solución de un problema de tipo *Permutation Flow Shop* o *Flexible Job Shop* estocástico.

8. Glosario

- **Flow shop:** Ambiente de programación en el que se cuenta con m máquinas situadas en serie y donde todos los trabajos deben pasar por cada una de ellas, de forma que cada una siga exactamente la misma ruta (Pinedo, 2016).
- **Flexible flow shop:** También llamado *hybrid flow shop* (HFS), es la generalización de un *flow shop* y de los entornos de máquinas en paralelo. En lugar de m máquinas en serie, hay c etapas en serie, en la que cada etapa tiene varias máquinas idénticas en paralelo. Cada trabajo debe ser procesado en la etapa 1, luego en la etapa 2, y así sucesivamente. Una etapa funciona como un banco de máquinas paralelas; en cada etapa, el trabajo j requiere procesamiento en una sola máquina y cualquier máquina puede hacerlo (Pinedo, 2016).
- **GRASP:** Es una metaheurística iterativa utilizada para problemas de optimización combinatorios, en donde cada iteración es realizada en dos fases: construcción y búsqueda local (Rajkumar, Asokan, Anilkumar, & Page, 2010).
- **Makespan:** Siendo C_j el tiempo de terminación del trabajo j , el makespan se define como el $\max(C_1, \dots, C_n)$. Así, es equivalente al tiempo de terminación del último trabajo en abandonar el sistema (Pinedo, 2016).
- **Metaheurística:** Anghinolfi & Paolucci (2007) definen una metaheurística como un conjunto de principios que guían el desarrollo de algoritmos para abordar problemas específicos; por lo tanto, los enfoques metaheurísticos generalmente requieren una fase de diseño para identificar características, estrategias y valores de parámetros que proporcionan los algoritmos con un comportamiento adecuado.
- **Preemptions:** Es una interrupción del procesamiento de una tarea. La cual implica que no es necesario mantener un trabajo en una máquina, una vez iniciado, hasta su terminación (Pinedo, 2016).
- **Robustez:** Capacidad de prevenir efectos indeseables resultantes de aproximaciones realizadas con un grado de incertidumbre -zona de ignorancia- (Roy, 2010).

- **Simheurística (*simheuristic*):** En el trabajo publicado por Juan, Faulin, Grasman, Rabe, & Figueira (2015) se define como una integración de simulación (en cualquiera de sus variantes) en un marco de metaheurística para resolver problemas complejos de optimización estocástica.
- **Simulación de Monte Carlo:** Es una técnica muy utilizada en el análisis probabilístico. Esta técnica de experimentación numérica se utiliza para obtener estadísticas de variables de entrada. En cada ejecución de la simulación, los valores de las variables aleatorias se basan en una distribución y los valores de las variables de salida se calculan utilizando un modelo computacional (Cruse, 1997).
- **Tardanza total:** La tardanza total es la suma de la tardanza de todos los trabajos programados, es decir, $T = \sum T_j$. En los problemas de programación, es un objetivo de minimización (Baker, 2013).
- **Tiempo total de terminación (*total completion time*):** También conocido como *flowtime*, es la suma de los tiempos de terminación de los trabajos en el ambiente de producción (Adiri, Bruno, Kan, & Rinnooy, 1989).

Referencias

- Adiri, I., Bruno, J., Kan, E. F., & Rinnooy, A. H. G. (1989). Single machine flow-time scheduling with a single breakdown. *Acta Informatica*, 26(7), 679–696.
<https://doi.org/https://doi.org/10.1007/BF00288977>
- Almeder, C., & Hartl, R. F. (2012). A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. *International Journal of Production Economics*, 145(1), 88–95.
<https://doi.org/10.1016/j.ijpe.2012.09.014>
- Anghinolfi, D., & Paolucci, M. (2007). Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach. *Computers and Operations Research*, 34(11), 3471–3490.
<https://doi.org/10.1016/j.cor.2006.02.009>
- Baker, K., & Althamer, D. (2012). Heuristic solution methods for the stochastic flow shop problem. *European Journal of Operational Research*, 216(1), 172–177.
<https://doi.org/10.1016/j.ejor.2011.07.021>
- Baker, K. R. (1974). *Introduction to sequencing and scheduling*. New York: Wiley.
- Baker, K. R. (2013). Computational results for the flowshop tardiness problem. *Computers and Industrial Engineering*, 64(3), 812–816. <https://doi.org/http://doi.org/10.1016/j.cie.2012.12.018>
- Banerjee, B. P. (1965). Single Facility Sequencing with Random Execution Times. *Operations Research*, 13(3), 358–364. <https://doi.org/https://doi.org/10.1287/opre.13.3.358>
- Brah, S. A., & Wheeler, G. E. (1998). Comparison of Scheduling Rules in a Flow Shop with Multiple Processors: A Simulation. *Simulation*, 71(5), 302–311. <https://doi.org/10.1177/003754979807100501>
- Choi, S. H., & Wang, K. (2012). Flexible flow shop scheduling with stochastic processing times: A decomposition-based approach. *Computers and Industrial Engineering*, 63(2), 362–373.
<https://doi.org/10.1016/j.cie.2012.04.001>
- Croes, G. A. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6), 791–812. <https://doi.org/10.1287/opre.6.6.791>
- Cruse, T. A. (1997). *Reliability based mechanical design*. Nashville: Marcel Dekker Inc.
- Dugardin, F., Yalaoui, F., & Amodeo, L. (2010). New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *European Journal of Operational Research*, 203(1), 22–31.
<https://doi.org/http://doi.org/10.1016/j.ejor.2009.06.031>
- Elyasi, A., & Salmasi, N. (2013a). Stochastic flow-shop scheduling with minimizing the expected number of tardy jobs. *International Journal of Advanced Manufacturing Technology*, 66(1–4), 337–346.
<https://doi.org/10.1007/s00170-012-4328-4>
- Elyasi, A., & Salmasi, N. (2013b). Stochastic scheduling with minimizing the number of tardy jobs using chance constrained programming. *Mathematical and Computer Modelling*, 57(5–6), 1154–1164.
<https://doi.org/10.1016/j.mcm.2012.10.017>
- Fazayeli, M., Aleagha, M. R., Bashirzadeh, R., & Shafaei, R. (2016). A hybrid meta-heuristic algorithm for flowshop robust scheduling under machine breakdown uncertainty. *International Journal of Computer Integrated Manufacturing*, 29(7), 709–719. <https://doi.org/10.1080/0951192X.2015.1067907>
- Feng, X., Zheng, F., & Xu, Y. (2016). Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *International Journal of Production Research*, 54(12), 3706–3717.
<https://doi.org/10.1080/00207543.2016.1162341>

- Feo, T. ., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2), 109–133. <https://doi.org/http://doi.org/10.1007/BF01096763>
- Ferone, D., Gruler, A., Festa, P., & Juan, A. A. (2016). Combining simulation with a GRASP metaheuristic for solving the permutation flow-shop problem with stochastic processing times. In *Proceedings of the 2016 Winter Simulation Conference* (pp. 2205–2215). <https://doi.org/10.1109/WSC.2016.7822262>
- Figueira, G., & Almada-Lobo, B. (2014). Hybrid simulation-optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46, 118–134. <https://doi.org/10.1016/j.simpat.2014.03.007>
- Framinan, J. M., & Leisten, R. (2003). An efficient constructive heuristic for flowtime minimisation in permutation flow shops. *Omega*, 31(4), 311–317. [https://doi.org/10.1016/S0305-0483\(03\)00047-1](https://doi.org/10.1016/S0305-0483(03)00047-1)
- Framinan, J. M., Leisten, R., & Ruiz-Usano, R. (2002). Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation. *European Journal of Operational Research*, 141(3), 559–569. [https://doi.org/10.1016/S0377-2217\(01\)00278-8](https://doi.org/10.1016/S0377-2217(01)00278-8)
- Framinan, J. M., Leisten, R., & Ruiz-Usano, R. (2005). Comparison of heuristics for flowtime minimisation in permutation flowshops. *Computers and Operations Research*, 32(5), 1237–1254. <https://doi.org/10.1016/j.cor.2003.11.002>
- Gholami, M., Zandieh, M., & Alem-Tabriz, A. (2009). Scheduling hybrid flow shop with sequence-dependent setup times and machines with random breakdowns. *International Journal of Advanced Manufacturing Technology*, 42(1–2), 189–201. <https://doi.org/10.1007/s00170-008-1577-3>
- Gonzalez-Neira, E. M., Ferone, D., Hatami, S., & Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *European Journal of Operational Research*, 79, 23–36. <https://doi.org/10.1016/j.simpat.2017.09.001>
- González-Neira, E. M., & Montoya-Torres, J. R. (2017). A GRASP meta-heuristic for the hybrid flowshop scheduling problem. *Journal of Decision Systems*, 26(3), 294–306. <https://doi.org/10.1080/12460125.2017.1351863>
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In *Annals of Discrete Mathematics* (Vol. 5, pp. 287–326). Elsevier. [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- Grasas, A., Juan, A. A., & Lourenço, H. R. (2014). SimLLS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1), 1–9. <https://doi.org/10.1057/jos.2014.25>
- Ji, M., Yang, Y., Duan, W., Wang, S., & Liu, B. (2016). Scheduling of no-wait stochastic distributed assembly flowshop by hybrid PSO. *2016 IEEE Congress on Evolutionary Computation*, (71101139), 2649–2654. <https://doi.org/10.1109/CEC.2016.7744120>
- Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46, 101–117. <https://doi.org/10.1016/j.simpat.2014.02.005>
- Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2, 62–72. <https://doi.org/10.1016/j.orp.2015.03.001>
- Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2, 62–72. <https://doi.org/http://doi.org/10.1016/j.orp.2015.03.001>
- Kim, Y.-D., Lim, H.-G., & Park, M.-W. (1996). Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. *European Journal of Operational Research*, 91(1), 124–143. [https://doi.org/10.1016/0377-2217\(95\)00119-0](https://doi.org/10.1016/0377-2217(95)00119-0)
- Kim, Y. D. (1993). Heuristics for flowshop scheduling problems minimizing mean tardiness. *Journal of the Operational Research Society*, 44(1), 19–28. <https://doi.org/10.1057/jors.1993.3>
- Knowles, J., & Corne, D. (1999). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 1, 98–105. <https://doi.org/10.1109/CEC.1999.781913>
- Knowles, J. D., & Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2), 149–172. <https://doi.org/10.1162/106365600568167>
- Kubzin, M. A., & Strusevich, V. A. (2004). Two-machine flow shop no-wait scheduling with machine maintenance. *4or*, 3(4), 303–313. <https://doi.org/10.1007/s10288-005-0070-1>

- Li, Z., & Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers and Chemical Engineering*, 32(4–5), 715–727. <https://doi.org/10.1016/j.compchemeng.2007.03.001>
- Lin, Y., Huang, D., & Lin, J. (2016). Reliability evaluation of a multistate flexible flow shop with stochastic capacity for multiple types of Jobs. *Journal of Manufacturing Systems*, 41, 287–298. <https://doi.org/https://doi.org/10.1016/j.jmsy.2016.10.003>
- Makino, T. (1965). On a scheduling problem. *Journal of the Operations Research Society Japan*, 8, 32–44.
- Nahmias, S. (2007). *Análisis de la producción y las operaciones* (5th Editio). Boston.
- Nawaz, M., Ensore, E. E., & Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1), 91–95. [https://doi.org/10.1016/0305-0483\(83\)90088-9](https://doi.org/10.1016/0305-0483(83)90088-9)
- Pan, Q.-K., & Ruiz, R. (2012). Local search methods for the flowshop scheduling problem with flowtime minimization. *European Journal of Operational Research*, 222(1), 31–43. <https://doi.org/10.1016/j.ejor.2012.04.034>
- Pan, Q. K., Ruiz, R., & Alfaro-Fernández, P. (2017). Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows. *Computers and Operations Research*, 80, 50–60. <https://doi.org/10.1016/j.cor.2016.11.022>
- Pinedo, M. L. (2016). *Scheduling. Theory, Algorithms and Systems* (5 th Editi). New York.
- Rabiee, M., Jolai, F., Asefi, H., Fattahi, P., & Lim, S. (2016). A biogeography-based optimisation algorithm for a realistic no-wait hybrid flow shop with unrelated parallel machines to minimise mean tardiness. *International Journal of Computer Integrated Manufacturing*, 29(9), 1007–1024. <https://doi.org/10.1080/0951192X.2015.1130256>
- Rahmani, D., Ramezani, R., & Mehrabad, M. S. (2014). Multi-objective flow shop scheduling problem with stochastic parameters: fuzzy goal programming approach. *International Journal of Operational Research*, 21(3), 322–340. <https://doi.org/https://doi.org/10.1504/IJOR.2014.065411>
- Rajkumar, M., Asokan, P., & Vamsikrishna, V. (2010). A GRASP algorithm for flexible job-shop scheduling with maintenance constraints. *International Journal of Production Research*, 48(22), 6821–6836. <https://doi.org/10.1080/00207540903308969>
- Ribas, I., Leisten, R., & Framiñan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers and Operations Research*, 37(8), 1439–1454. <https://doi.org/10.1016/j.cor.2009.11.001>
- Roy, B. (2010). Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3), 629–638. <https://doi.org/https://doi.org/10.1016/j.ejor.2008.12.036>
- Ruiz, R., & Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1–18. <https://doi.org/10.1016/j.ejor.2009.09.024>
- Tang-Liu, D., & Acheampong, A. (2005). Ocular pharmacokinetics and safety of ciclosporin, a novel topical treatment for dry eye. *Clinical Pharmacokinetics*, 44(3), 247–261. <https://doi.org/10.2165/00003088-200544030-00003>
- Vallada, E., Ruiz, R., & Minella, G. (2008). Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers and Operations Research*, 35(4), 1350–1373. <https://doi.org/10.1016/j.cor.2006.08.016>
- Wang, L., Zhang, L., & Zheng, D. Z. (2005). A class of hypothesis-test-based genetic algorithms for flow shop scheduling with stochastic processing time. *International Journal of Advanced Manufacturing Technology*, 25(11–12), 1157–1163. <https://doi.org/10.1007/s00170-003-1961-y>
- Wiers, V. (1997). *Human-computer interaction in production scheduling: Analysis and design of decision support systems for production scheduling tasks*. Universidad técnica Eindhoven. Holanda.
- Yenisey, M. M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, 45, 119–135. <https://doi.org/10.1016/j.omega.2013.07.004>