



SELECCIONES MATEMÁTICAS

Universidad Nacional de Trujillo

ISSN: 2411-1783 (Online)

2019; Vol.06(2):297 - 310.



REVIEW

Medidas de robustez y estabilidad para problemas de scheduling con incertidumbre: Una revisión del estado del arte.

Measures of robustness and stability to scheduling problems with uncertainty: A review of the state of the art.

Rafael Asmat U.*^{id}, Edmundo Vergara M.^{†id}, and Flabio Gutiérrez S.^{‡id}

Received, Oct. 01, 2019

Accepted, Nov. 28, 2019



How to cite this article:

Asmat, R., et al. *Medidas de robustez y estabilidad para problemas de scheduling con incertidumbre: Una revisión del estado del arte*. *Selecciones Matemáticas*. 2019; 6(2):297-310. <http://dx.doi.org/10.17268/sel.mat.2019.02.16>

Resumen

En este trabajo se resumen los diversos enfoques y políticas más usados en la literatura para medir la robustez y la estabilidad de soluciones en problemas de scheduling. Estas políticas se basan en análisis de problemas de scheduling proactivos con un enfoque de planeación de escenarios. Además, se presentan y revisan los estudios más recientes desarrollados para medir la robustez y la estabilidad de soluciones a problemas de scheduling con incertidumbre y se sugieren futuras líneas de investigación.

Palabras Clave. Robustez, Estabilidad, Problema de Scheduling, Incertidumbre, Reactivo, Proactivo.

Abstract

In this work we summarize the different and more used approaches and policies found in the literature to measure the robustness and stability of solutions for scheduling problems. These policies are based in the analysis of proactive scheduling problems with an approach of planning scenes. Moreover, we present and review the more recent studies developed to measure the robustness and the stability of solutions for uncertainty scheduling problems and we suggest future lines of research.

Keywords. Robustness, Stability, Scheduling Problem, Uncertainty, Reactive, Proactive.

1. Introducción. Los problemas de scheduling son de mucha importancia en la industria moderna, debido a lo cual han recibido la atención de un gran número de investigadores en diversas áreas [14, 18, 34]. Una característica inherente en este tipo de problemas es su incertidumbre, la cual está relacionada con las necesidades de cambio de los clientes, la estimación errónea en la duración de las tareas, la indisponibilidad temporal de recursos, etc. Entre los diversos estudios enfocados en contrarrestar los efectos de las interrupciones impredecibles en la producción, dos enfoques han adquirido un considerable interés en los investigadores: el *enfoque reactivo* y el *enfoque proactivo* [36].

El enfoque reactivo tiene por principal objetivo obtener una nueva solución bastante similar a la solución encontrada previamente, antes que ocurran los cambios y que sea lo más eficiente posible. El enfoque proactivo usa el conocimiento sobre los futuros cambios para evitar o minimizar sus efectos. En el proceso

*Departamento de Matemáticas, Universidad Nacional de Trujillo, Trujillo-Perú. (rasmat@unitru.edu.pe).

†Departamento de Matemáticas, Universidad Nacional de Trujillo, Trujillo-Perú. (evergara@unitru.edu.pe).

‡Departamento de Matemáticas, Universidad Nacional de Piura, Urb. Miraflores s/n, Piura-Perú. (flabio@unp.edu.pe).

de scheduling se utilizan ciertas medidas regulares (makespan, flowtime, tardanza, etc.) para medir el rendimiento del sistema. Sin embargo, la naturaleza dinámica y estocástica del proceso de scheduling, genera interrupciones imprevistas que afectan a la medida de rendimiento seleccionada y los schedules generados ya no son factibles. Es por ello que el schedule inicial debe revisarse parcial o globalmente para adaptarse a estas interrupciones y mantener su factibilidad generando un nuevo schedule llamado realizado.

Para tratar la incertidumbre se utilizan actualmente dos nuevos criterios: la robustez y la estabilidad. A pesar que algunos autores utilizan ambos para dar a entender los mismos conceptos [29], éstos deben distinguirse de forma clara, pues representan diferentes comportamientos de la solución del problema luego de los cambios en el entorno. Un schedule cuyo rendimiento no se deteriora significativamente debido a las interrupciones es llamado robusto y un schedule cuyos eventos realizados no se desvían significativamente del schedule original es llamado estable [37].

En este trabajo esclarecemos la diferencia entre la robustez y la estabilidad y presentamos los métodos más importantes para determinar ambos criterios en problemas de scheduling. El trabajo está organizado de la siguiente manera: En la sección 2 nos enfocamos en los conceptos de robustez y estabilidad para problemas de scheduling con incertidumbre; proporcionamos una definición más clara para entender la diferencia entre una solución robusta y una solución estable. En las secciones 3 y 4 presentamos las diversas medidas y políticas de robustez y estabilidad usadas en la literatura. En la sección 5 revisamos los estudios recientes en la literatura sobre robustez y estabilidad. Finalmente presentamos las conclusiones en la sección 6.

2. Robustez y Estabilidad en Problemas de Scheduling. En un proceso de scheduling se genera primero un schedule llamado inicial, en el cual se orientan las futuras actividades de planta. Sin embargo, debido al entorno dinámico y estocástico de los problemas de scheduling, se presentan interrupciones imprevistas: avería de máquinas, cancelaciones de órdenes, cambio en los plazos, variaciones en los tiempos de procesamiento, etc. Estas interrupciones evitan la ejecución del schedule planeado originalmente. Debido a ello, este schedule debe ser revisado parcial o globalmente para mantener su factibilidad. El schedule que se ejecuta realmente en la planta es llamado schedule realizado y puede diferir sustancialmente del schedule inicial. La robustez y la estabilidad están relacionadas con esta diferencia.

2.1. Robustez. De modo general, se define la robustez como la capacidad de un sistema para soportar perturbaciones, presiones o cambios impredecibles en su entorno operacional sin pérdida de funcionalidad (Barber y Salido, [3]). En un problema de scheduling, el término robustez se refiere a la insensibilidad en el rendimiento del schedule frente a las interrupciones. Un schedule cuyo rendimiento no se deteriora demasiado frente a las interrupciones es llamado robusto. El principal interés será entonces en el rendimiento del schedule realizado (Goren y Sabuncuoglu, [11]), lo cual convierte a la robustez en una medida de rendimiento práctica.

2.2. Estabilidad. La estabilidad es un concepto derivado de la astronomía y la física (Wiggins, [38]). Se dice que la solución de un sistema dinámico (un estado de equilibrio) es *estable* si pequeñas perturbaciones en la solución generan una nueva solución “cercana” a la solución original. Estas perturbaciones son pequeñas diferencias ocurridas en el estado actual del sistema.

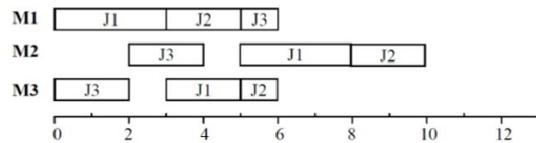
En un problema de scheduling, la estabilidad se refiere a la diferencia entre el schedule inicial y el schedule realizado. Un schedule cuya realización no se desvía demasiado del schedule inicial, frente a las interrupciones, es llamado *estable* (Barber y Salido, [3]).

Un schedule sirve como plan general para otras actividades de producción de planta: determinación de las fechas de entrega, los tiempos de finalización, los requisitos para la planificación de recursos secundarios, etc. Por lo tanto, cualquier desviación de la producción en el schedule puede interrumpir dichas actividades secundarias y aumentar el nerviosismo del sistema. Así, la estabilidad también es una medida importante (Goren y Sabuncuoglu, [11]).

La robustez y la estabilidad se ilustran mejor con la ayuda del cuadro de Gantt dado por la figura 1.

La figura (a) muestra un schedule inicial para un problema de job shop con 3 máquinas (M_1 , M_2 y M_3) y 3 trabajos (J_1 , J_2 y J_3) sujeto a interrupciones aleatorias y la figura (b), su posible realización. El área sombreada en el schedule realizado (b) de la máquina 2 entre los tiempos 7 y 9 representa una interrupción. Si la medida de rendimiento es el máximo tiempo de completamiento (C_{max}), entonces, para medir la robustez nos centramos en el rendimiento del schedule realizado ($C_{max} = 13$) definiendo una medida de robustez a optimizarse (por ejemplo, C_{max} esperado), es decir, que minimice la desviación del rendimiento entre ambos schedules (aquí, $C_{max} = 13 - 10 = 3$). Por otro lado, la operación del trabajo 1 en la máquina 2 se completa después de lo planeado, afectando el procesamiento del trabajo 2 en esa máquina, planeado originalmente entre los tiempos 8 y 10. Al tomar en cuenta la estabilidad, ese tipo de desviaciones del schedule inicial deben minimizarse. Por tanto, definimos una medida de estabilidad (por ejemplo, la suma esperada de las desviaciones absolutas en los tiempos de completamiento) que minimice esas desviaciones

PLANEADO:



REALIZADO:

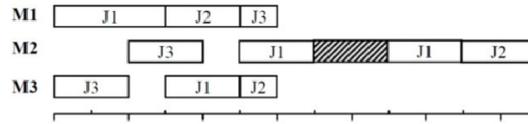


FIGURA 2.1. Un schedule inicial (a) y su realización (b)

entre ambos schedules. (Goren y Sabuncuoglu, [10]).

3. Enfoques y Medidas de Robustez y Estabilidad. El principal objetivo del proceso de scheduling es conseguir un buen rendimiento de los schedules realizados sin que estos se desvíen significativamente de los iniciales. Para lograr estos objetivos y afrontar la incertidumbre, se consideran dos políticas: *scheduling reactivo* y *scheduling proactivo*.

El *scheduling reactivo* no considera directamente la incertidumbre en la generación de schedules, sino más bien revisa el schedule durante la ocurrencia de interrupciones o eventos inesperados. El objetivo principal al reaccionar a una interrupción es, por lo general, minimizar la desviación entre el nuevo schedule y el inicial. También puede considerarse el minimizar la degradación en la medida de rendimiento (Wu et al, [40]).

El *scheduling proactivo*, por otro lado, considera las futuras interrupciones al momento de generar los schedules iniciales de modo que se minimicen los efectos de las mismas en las medidas de rendimiento. Aunque el interés primario en este enfoque es la robustez, también puede considerarse la estabilidad (Mehta y Uzsoy [26, 27]).

Usar un enfoque reactivo implica volver a resolver el problema luego de la pérdida de cada solución, lo cual es una gran desventaja sobre todo cuando trabajamos con cambios a corto plazo. Como el principal interés es evitar la pérdida de soluciones en este tipo de entornos, la gran mayoría de los enfoques se concentran en los schedules proactivos, dirigidos a la búsqueda de soluciones robustas y estables.

3.1. Scheduling Proactivo. En esta sección nos enfocaremos en los scheduling proactivos. El principal objetivo es encontrar un schedule bueno, aunque no necesariamente óptimo, que soporte futuras interrupciones imprevistas generadas por las propias condiciones dinámicas y estocásticas de los problemas de scheduling. A continuación, presentamos las diversas formas para generar estos schedules así como las diversas medidas de rendimiento consideradas.

3.1.1. Notaciones y Términos. Consideraremos la notación utilizada por Sabuncuoglu y Goren, [30]. Dado un problema de scheduling inicial P , con m soluciones (schedules) factibles, S_1, \dots, S_m , deseamos encontrar un schedule óptimo S_0^* de acuerdo a la medida de rendimiento seleccionada. Si $f(S)$ es el valor de esta medida para el schedule S , entonces seleccionamos el schedule S_0^* que minimice esta medida de rendimiento, es decir, que satisface

$$f(S_0^*) \leq f(S_i), \quad i = 1, \dots, m. \tag{3.1}$$

En la figura 2 tenemos n escenarios correspondientes a posibles interrupciones, y un escenario P_0 que representa ninguna interrupción. Por tanto, tenemos $n + 1$ escenarios en total. El problema original P_0 cambia por el problema P_j para cada escenario j con probabilidad a_j . En consecuencia, el schedule S_i cambia por el schedule S_{ij} en el j -ésimo escenario. Esto quiere decir que, S_{ij} es la versión realizada del schedule inicial S_i al ocurrir la interrupción en el escenario j . Denotemos por S_j^* a la solución óptima del problema P_j y $f(S_j^*)$ al valor de la función objetivo.

4. Políticas y Medidas de Robustez y Estabilidad. Esta sección resume las diversas medidas de robustez o estabilidad discutidas en la literatura actual generadas mediante el uso de los principios existentes de la teoría de toma de decisiones.

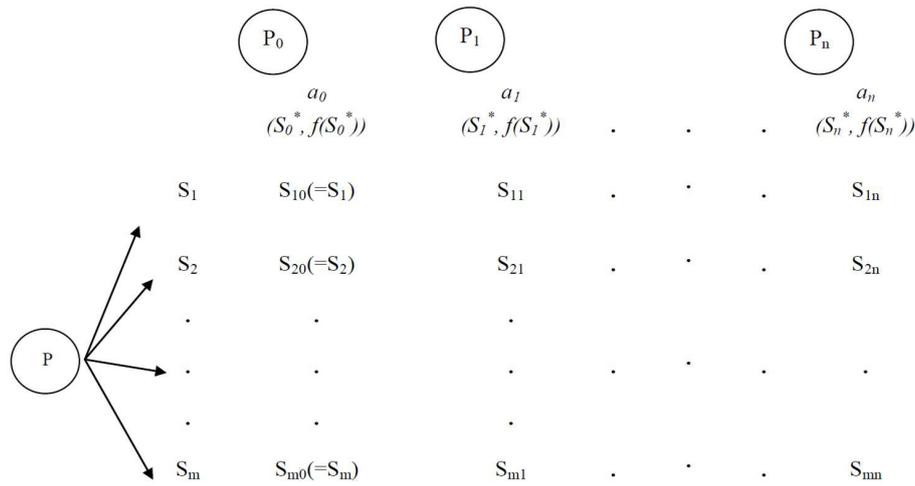


FIGURA 3.1. Escenario de representación de las interrupciones

4.1. Medidas de Robustez. Algunas medidas de robustez se basan en el rendimiento actual de los schedules realizados, $f(S_{ij})$ y otras se basan en *regrets*. El *regret* asociado con el schedule inicial S_i y el escenario j es la diferencia entre los rendimientos óptimos y realizados, i.e., $f(S_{ij}) - f(S_j^*)$. El primero tiene como objetivo seleccionar el schedule con un buen rendimiento realizado, mientras que el segundo intenta seleccionar un schedule cuyo rendimiento no sea relativamente malo con respecto al mejor rendimiento.

4.1.1. Medidas de Robustez Basadas en Rendimientos Realizados.

1. Minimizar el rendimiento realizado esperado

$$S^* = S_t, t \in \arg \min_i \{E[f(S_i)]\}, \tag{4.1}$$

donde $E[f(S_i)] = \sum_{j=0}^n a_j f(S_{ij})$. Este método selecciona el schedule cuya medida de rendimiento es, en promedio, la mejor. Es el más usado en la literatura basado en la medida de robustez (Wu et. al., [40]).

2. Minimizar el rendimiento de peor caso

$$S^* = S_t, t \in \arg \min_i \{\max_j \{f(S_{ij})\}\}. \tag{4.2}$$

Esta política selecciona el schedule cuya medida de rendimiento es, en el peor de los casos, mejor que la de otros (Artigues et. al., [2].)

3. Minimizar el rendimiento del escenario de peor caso, cuando éste es conocido

$$S^* = S_t, t \in \arg \min_i \{f(S_{ik})\}, \tag{4.3}$$

donde k es el escenario de peor caso.

4. Minimizar el rendimiento del escenario más probable

$$S^* = S_t, t \in \arg \min_i \{f(S_{im})\}, \tag{4.4}$$

donde $m \in \arg \max_j \{a_j\}$.

5. Seleccionar un schedule de modo que se minimize la desviación esperada del rendimiento del schedule realizado desde el rendimiento determinístico inicial

$$S^* = S_t, t \in \arg \min_i \{\sigma_i\}, \tag{4.5}$$

donde $\sigma_i = |f(S_{i0}) - E[f(S_i)]|$. Esta política enfatiza la definición: “El schedule robusto es aquel cuyo rendimiento se degrada mínimamente de cara a las interrupciones”(Leon et. al. [25]).

6. Minimizar la varianza de la medida de rendimiento realizada

$$S^* = S_t, t \in \arg \min_i \{Var[f(S_i)]\}, \tag{4.6}$$

donde,

$$Var[f(S_i)] = \sum_{j=0}^n a_j f^2(S_{ij}) - (\sum_{j=0}^n a_j f(S_{ij}))^2.$$

El principio de esta política es encontrar el schedule cuya degradación de rendimiento sea mínimo de cara a las interrupciones, como en la medida anterior (Sevaux y Sorensen, [32]).

7. Seleccionar el schedule que minimice una medida que sea una combinación convexa de las medidas antes mencionadas. Por ejemplo, seleccionar el schedule que minimice $rE[f(S_i^r)] + (1-r)\sigma_i$, donde r es un número real entre 0 y 1. La primera parte de esta medida enfatiza el hecho de que un schedule robusto debería funcionar bien frente a las interrupciones. La segunda parte enfatiza: el rendimiento de un schedule robusto no debería degradarse mucho frente a las interrupciones. Al variar r entre 0 y 1, se pueden asignar diferentes pesos a estos dos aspectos de robustez (Leon et al. [25]).

4.1.2. Medidas de Robustez Basadas en Regrets. Al considerar las pérdidas de oportunidades o regrets, podemos comparar el rendimiento realizado con el rendimiento óptimo que deberíamos haber obtenido, al tener disponible la información adecuada. Primero construimos la matriz de regrets, $\Delta = [\delta_{ij}]$, donde $\delta_{ij} = |f(S_{ij}) - f(S_j^*)|$, $i = 1, \dots, m$ y $j = 1, \dots, n$. El ij -ésimo elemento de esta matriz es la diferencia entre el rendimiento óptimo del j -ésimo escenario y el rendimiento realizado del i -ésimo schedule en este escenario. Las medidas de robustez más usuales basadas en regrets son las siguientes:

1. Minimizar el regret esperado

$$S^* = S_t, t \in \arg \min_i \{E[\delta_i]\}, \tag{4.7}$$

donde $E[\delta_i] = \sum_{j=0}^n a_j \delta_{ij}$. Este método selecciona un schedule cuyo regret es, en promedio, el menor.

2. Minimizar el regret de peor caso

$$S^* = S_t, t \in \arg \min_i \{\max_j \{\delta_{ij}\}\}. \tag{4.8}$$

Es el más usado en la literatura basado en medidas de robustez (Daniels y Kouvelis [7]).

3. Minimizar el escenario de regret de peor caso

$$S^* = S_t, t \in \arg \min_i \{\delta_{ik}\}, \tag{4.9}$$

donde k es el escenario de peor caso. Esta política selecciona S_i con $\min_i \{\delta_{ik}\}$, donde P_k es el escenario de peor caso.

4. Minimizar el escenario de regret más probable

$$S^* = S_t, t \in \arg \min_i \{\delta_{im}\}, \tag{4.10}$$

donde $m \in \arg \max_j \{a_j\}$.

4.2. Medidas de Estabilidad. Las medidas de estabilidad se basan, principalmente, en las diferencias entre los tiempos de inicio/finalización de las operaciones, el número operaciones interrumpidas o el número de cambios en el scheduling. Sean C_k^S el tiempo de finalización del trabajo k mediante el schedule S y N el número de trabajos. Construimos la matriz de diferencias de los tiempos de finalización $D = [d_{ij}]$, donde $d_{ij} = \sum_{k=1}^N |C_k^{S_{ij}} - C_k^{S_{i0}}|$. Las medidas de estabilidad consideradas son las siguientes:

1. Minimizar las diferencias esperadas

$$S^* = S_t, t \in \arg \min_i \{E[d_i]\}, \tag{4.11}$$

donde $E[d_i] = \sum_{j=0}^n a_j d_{ij}$. Este método selecciona un schedule cuya estabilidad es, en promedio, la mejor. Es la más utilizada en la literatura ([26, 27, 28]).

2. Minimizar el cuadrado de la diferencia esperada

$$S^* = S_t, t = \arg \min_i E[d_i^2], \quad (4.12)$$

donde $E[d_i^2] = \sum_{j=0}^n a_j d_{ij}^2$. Este método selecciona un schedule cuya estabilidad es, en promedio, la mejor y es análoga a la medida “error promedio al cuadrado” usada en estadística.

3. Minimizar la varianza total de los tiempos de finalización realizados

$$S^* = S_t, t = \arg \min_i \sum_{k=1}^N \text{Var}[C_k^{r_i}], \quad (4.13)$$

donde $C_k^{r_i}$ es el tiempo de finalización realizado del trabajo k bajo el schedule i . Por tanto, $\text{Var}[C_k^{r_i}] = \sum_{j=1}^n a_j (C_k^{S_{ij}})^2 - (\sum_{j=1}^n a_j C_k^{S_{ij}})^2$.

4. Minimizar la diferencia de tiempo de finalización de peor caso

$$S^* = S_t, t \in \arg \min_i \{ \max_j \{ d_{ij} \} \}. \quad (4.14)$$

Esta política selecciona un schedule cuya estabilidad en el peor de los casos es mayor que las demás.

5. Minimizar la diferencia de tiempo de finalización en el escenario de peor caso

$$S^* = S_t, t \in \arg \min_i \{ d_{ik} \}, \quad (4.15)$$

donde k es el escenario de peor caso.

6. Minimizar la diferencia de tiempo de finalización del escenario más probable

$$S^* = S_t, t \in \arg \min_i \{ d_{im} \}, \quad (4.16)$$

donde $m \in \arg \max_j \{ a_j \}$.

El costo computacional para calcular estas medidas de robustez es bastante alto debido a que los escenarios no pueden definirse de antemano con tanta facilidad y porque, además, su número es potencialmente infinito. Es por ello que los trabajos actuales se enfocan en el uso de las llamadas medidas sustitutas, medidas auxiliares fáciles de calcular y que están fuertemente correlacionadas con las medidas usuales. Otros autores dirigen sus estudios hacia la simulación para poder estimar los valores de las medidas de robustez y/o estabilidad.

5. Estudios Recientes sobre Robustez y Estabilidad en Scheduling. En esta sección presentamos los principales estudios presentes en la literatura sobre robustez y estabilidad en problemas de scheduling con incertidumbre, es donde se definen y optimizan ciertas medidas. La mayoría de estos estudios son proactivos y se dividen en dos categorías: Enfoques probabilísticos (modelan la incertidumbre mediante funciones de densidad probabilísticas y las medidas de robustez o estabilidad se definen en términos de estas funciones) y Enfoques de planeación de escenarios (proactivos). Ambos enfoques son análogos pues el enfoque probabilístico puede considerarse como un enfoque de planificación con un continuo de infinitos escenarios y la probabilidad para cada escenario está gobernada por las funciones de densidad probabilísticas. Por tanto, las medidas de robustez o de estabilidad usadas en ambos enfoques son análogas [30].

5.1. Enfoque Probabilístico.

5.1.1. Estudios sobre Robustez.

- Leon et al. [25] estudian la robustez en un entorno job-shop con el objetivo de construir un schedule inicial robusto. Las interrupciones son averías de máquinas y las distribuciones para los tiempos de las interrupciones y reparaciones son conocidos. La medida de rendimiento es el makespan y la medida de robustez de un schedule toma como una combinación convexa del makespan esperado en el schedule realizado y la desviación esperada del makespan original determinista. Proponen varias medidas sustitutas y luego determinan la mejor de todas. Posteriormente utilizan un algoritmo genético para minimizar esta medida sustituta. El rendimiento del algoritmo propuesto se verifica al considerar avería aleatoria de máquina y procesar la variabilidad de tiempo. Sus resultados muestran que el algoritmo propuesto mejora a los clásicos algoritmos que minimizan sólo el makespan.

- Li y Glazebrook, 1998 [22] consideran el problema de scheduling en un entorno de single machine con tiempos de procesamiento aleatorios, múltiples averías de máquina y una política de anticipación-reanudación para minimizar la suma ponderada de una función creciente en los tiempos de finalización esperados. Los autores desarrollan una regla de dominancia, basada en el intercambio por pares de los trabajos adyacentes, la cual se puede relajar para permitir que los tiempos de producción se distribuyan como una mezcla de exponenciales y de acuerdo una distribución Gamma. Sin embargo, esta regla de dominancia no puede aplicarse a medidas relacionadas con los plazos de vencimiento.
- Wu et al, 1999, [39] utilizan una descomposición gráfico-teórica en un problema de scheduling job shop para obtener la robustez del schedule. Su medida de robustez es el retraso ponderado promedio esperado. Los autores proponen un algoritmo branch and bound que procesa los arcos disyuntivos que representan las restricciones de precedencia y cambia algunos de ellos a conjuntivos. El resto de las decisiones se efectúan de manera dinámica aplicando la heurística Costo Aparente de la Tardanza (ATC). Este enfoque es llamado el esquema primero procesar, después programar (PFSL). Sus experimentos computacionales muestran que el esquema PFSL produce rendimientos más robustos frente un amplio rango de perturbaciones.
- Jensen, 2001, [15] analiza el rendimiento de las medidas de robustez basadas en vecindades, en un entorno job shop, sujeto a interrupciones. Las medidas de rendimiento consideradas son: tardanza máxima, tardanza total o flowtime total. En los problemas con tardanza, el autor también propone minimizar una medida de retraso, en vez de una de tardanza, para poder incrementar la holgura en el schedule y con ello mejorar el rendimiento del re-scheduling. El autor compara el rendimiento de las medidas de robustez basadas en vecindades con las medidas de robustez simples y con un criterio de minimización de la tardanza. Sus experimentos computacionales indican que las medidas de robustez basadas en retrasos mejoran la robustez del schedule en problemas con pérdidas de plazos de entrega y que éstas son equivalentes al scheduling ordinario en problemas estrictos. Estas medidas de robustez basadas en vecindades mejoran la robustez del schedule al usarse con técnicas de re-scheduling en todos los casos y esta mejora supera a la obtenida por la medida de robustez basada en retrasos, en la mayoría de los casos.
- Jensen, 2003, [16], utiliza como medida de rendimiento al makespan. El autor define dos medidas de robustez basadas en vecindades: el makespan promedio de la vecindad del schedule dado y una medida estimada (cota superior) de la primera. En la primera medida, la vecindad se define como el conjunto de todos los schedules que pueden obtenerse mediante un intercambio por pares de dos operaciones consecutivas en la misma máquina. Se comparan los rendimientos de estas dos medidas con los criterios "minimizar el makespan" "maximizar la holgura". Se utilizan algoritmos genéticos específicos y sus rendimientos se comparan bajo cinco diferentes métodos de re-scheduling. Los resultados computacionales indican que es posible encontrar un schedule ya sea con un makespan bajo o una medida de robustez baja. Se observa asimismo que, con un tiempo de interrupción moderado, la medida de robustez basada en holguras funciona muy bien en los métodos de re-scheduling simples, mientras que la robustez basada en vecindades supera a la medida basada en holguras en métodos de re-scheduling más sofisticados. Cuando se tienen tiempos de interrupción extensos, los resultados computacionales indican que la medida de robustez basada en holguras es superior en la mayoría de casos.
- Al-Fawzan y Haouari, 2005, [1], realizan un estudio en el contexto de scheduling con recursos limitados, donde la duración de las actividades está sujeta a incertidumbre. Los autores desarrollan un modelo con dos criterios para generar schedules robustos con un makespan mínimo. La robustez del schedule es la suma de las holguras libres de las actividades, las cuales se definen como la máxima cantidad de tiempo en que la actividad puede prolongarse sin retrasar el inicio de la siguiente actividad, manteniendo la factibilidad del recurso. El algoritmo propuesto es de búsqueda tabú, el cual aproxima el conjunto de soluciones eficientes. Los schedules se presentan como una lista ordenada de actividades y en cada iteración del algoritmo, la vecindad de una solución se genera mediante un intercambio por pares factible aleatorio de las actividades adyacentes. Los resultados computacionales indican que el algoritmo propuesto, en la mayoría de los casos, genera un conjunto de tamaño moderado de soluciones eficientes.
- Artigues, et al, 2005, [2] generan una familia de schedules para mantener la robustez del schedule en un entorno job shop. Esta familia de schedules se representa por asignaciones de grupos ordenados que definen una sucesión de grupos para cada máquina, de modo que las operaciones en cada grupo sean totalmente permutables. Este enfoque introduce el concepto de *flexibilidad*: en caso de interrupciones, los decisores pueden cambiar fácilmente de una solución a otra dentro de esa familia de schedules. Se proponen dos criterios para maximizar la flexibilidad: minimizar el

número de grupos y maximizar el número de schedules representados por esos grupos. El estudio se realiza, primeramente, en un escenario de single machine con plazos de entrega. Si todos los trabajos están listos en un tiempo $t = 0$, el algoritmo propuesto para minimizar el número de grupos es de tiempo polinomial. Si los plazos de lanzamiento de los trabajos no son todos cero, se propone un algoritmo de tiempo polinomial de programación dinámica para maximizar el número de schedules, el cual se usa como base para una heurística que genera asignaciones de grupos ordenados para problemas job shop con una Heurística de tipo Shifting Bottleneck. Sus resultados computacionales indican que el algoritmo propuesto incrementa la aplicabilidad del schedule dado, introduciendo una flexibilidad significativa manteniendo aceptable el makespan de los schedules representados por estas asignaciones.

- Goren y Sabuncuoglu, 2004, [31], realizan un enfoque de scheduling proactivo en un entorno de single machine con dos tipos de incertidumbre: avería de máquina y variabilidad en el tiempo de procesamiento. Definen dos medidas de robustez: flowtime total esperado y tardanza total esperada. Para una de las medidas de robustez desarrollan una regla de dominancia y dos cotas inferiores, utilizadas en un algoritmo de tipo branch and bound. También proponen un algoritmo heurístico Beam Search para resolver problemas de mayor tamaño en todas las medidas utilizadas. Sus resultados computacionales muestran que la heurística beam search genera schedules robustos con una pequeña desviación media del valor óptimo de la función objetivo y su rendimiento es mucho mejor que un número considerable de heurísticas disponibles en la literatura para todas las medidas consideradas.
- Leung y Pinedo, 2004, [23], estudian el problema de scheduling preemptivo en un entorno de parallel machine, sujeto a interrupciones aleatorias, tiempos de procesamiento determinísticos y plazos de entrega. Los autores desarrollan las condiciones en un número disponible de máquinas $m(t)$ bajo las cuales, diversas reglas minimizan el tiempo de finalización total, el makespan o la tardanza máxima. También analizan los casos donde existen plazos límites y restricciones de precedencia.
- Sevauy y Sorensen, 2004, [32], estudian la robustez en un entorno de single machine con plazos de liberación estocásticos. La medida de rendimiento usada es el número total ponderado de trabajos tardíos. Se desarrolla un algoritmo genético en donde el rendimiento realizado esperado y la desviación estándar del rendimiento realizado se emplean como funciones fitness para generar schedules robustos. Sus experimentos computacionales muestran que las funciones fitness robustas mejoran la robustez del schedule en comparación con el uso de la función fitness determinista.
- Chtourou y Haouari, 2008, [4], realizan un estudio de scheduling con recursos limitados considerando dos criterios de rendimiento: minimizar el makespan y maximizar una medida de robustez predictiva. El objetivo es seleccionar de un conjunto de schedules que comparte un makespan “bueno” aquel que ofrece la mejor robustez intrínseca. Se proponen 12 medidas de robustez sustitutas basadas en la noción de holgura libre. El algoritmo propuesto es una versión mejorada del serial SGS (esquema de generación de schedules) basado en un enfoque de generación de schedules de dos etapas. Sus resultados muestran los beneficios de elegir un schedule robusto sobre otros no robustos pero con el mismo makespan. Los schedules robustos mostraron un makespan menor en el 55 % de las veces, mientras que, los schedules no robustos tuvieron un mejor rendimiento sólo el 6 % de las veces.
- Barber y Salido, 2015, [3], se enfocan en un cierto tipo de problemas denominados CSP (problemas con satisfacción de restricciones), definiendo una medida de r -robustez variando de 0 a 1. Esta representa la probabilidad de que una solución continúe siendo solución del nuevo problema luego de la ocurrencia de un conjunto de incidencias Z . Mientras mayor sea su r -robustez, más robusta es una solución y mayor su probabilidad de permanecer factible luego de Z . Los autores aplican su medida de robustez a un problema de scheduling con satisfacción de restricciones temporales (TCSP) de tipo flow-shop. Sus resultados muestran que al ganar robustez para una solución del TCSP se pierde la optimalidad.
- Liao y van Wee, 2017, [24] proponen una familia de medidas de accesibilidad para expresar la robustez del sistema de transporte. Estas medidas expresan el número de opciones de viaje entre un origen y un destino dados, o actividades de conducta. Esta familia de medidas se conceptualiza usando el enfoque de múltiples estado llamado supernetwork. Con respecto a las opciones de viaje los autores discuten (1) el solapamiento de rutas de viaje y (2) la elección de los modos de viaje (por ejemplo, carro vs tren; multimodal). Con respecto a las actividades, discuten (1) la elección de los pares OD en contraposición a los patrones actividad-viaje y (2) ubicaciones de actividad fijas vs flexibles y orden de actividades fijas vs actividades flexibles. Además, ilustran las opciones de viaje y la robustez en un sistema de transporte multimodal entre los centros del corredor de la

Haya- Rotterdam-Dordrecht (Holanda). Los autores concluyen que un alto valor de robustez puede resultar de múltiples rutas de viaje con poca superposición a un conjunto estrecho de destinos, o de un conjunto limitado de rutas de viaje a una gama más amplia de destinos. Esto último es posiblemente más importante para los individuos que el primero.

5.1.2. Estudios Sobre Estabilidad.

- Wu et al., 1993, [40], estudian el problema de re-scheduling en un entorno de single-machine con avería de máquina reprogramando los trabajos para obtener un makespan mínimo y una estabilidad máxima. Ellos consideran dos criterios para medir la estabilidad: 1) La desviación del schedule revisado con respecto a los tiempos de inicio de los trabajos y 2) La desviación del schedule revisado. La medida para el primer criterio es la desviación absoluta promedio de los tiempos de inicio del trabajo original. El segundo criterio es difícil de calcular y por ello proponen una medida sustituta dada por la suma de las desviaciones absolutas de los tiempos de inicio de los trabajos del schedule *right-shift*. Se tienen dos tipos de problemas, los cuales difieren con respecto al criterio de estabilidad en uso, y se utilizan dos tipos de heurística para resolverlos; la primera basada en un intercambio por pares y la segunda con el uso de algoritmos genéticos. Realizaron dos grupos de experimentos generando primero pequeños problemas cuyas soluciones se compararon con la solución óptima y luego se miden la efectividad de la heurística con problemas grandes. En el primer experimento verifican que todas las heurísticas encontraban soluciones óptimas. En el segundo, confirman que la estabilidad de los schedules podía mejorarse significativamente sacrificando un poco el makespan.
- Sotskov et al, 1997, [33], introducen otro punto de vista para la estabilidad al manipular la incertidumbre en un entorno job shop con un análisis a posteriori, en el cual ya se obtuvo un schedule óptimo. Con esto se determina la variación máxima en el tiempo de procesamiento de las operaciones para que el schedule óptimo obtenido continúe siéndolo. Esa variación máxima es llamada el radio de “estabilidad” del schedule. Este tipo de análisis post optimalidad puede proporcionar algunas percepciones considerables sobre el impacto de la incertidumbre, pero también está asociado con algunos problemas: Si el “radio de estabilidad” del schedule óptimo es lo suficientemente grande para adaptarse a todos los posibles cambios en los tiempos de procesamientos, puede utilizarse con total seguridad el schedule óptimo obtenido, pero si no lo es, no es posible responder a la pregunta sobre qué curso de acción tomar.
- Mehta y Uzsoy, 1998, 1999, [26, 27], generan schedules iniciales *estables* con averías aleatorias de máquinas. Estos schedules fueron llamados *predecibles*, siendo este término un enfoque alternativo de lo que se conoce como scheduling/rescheduling. El objetivo de este enfoque es generar un schedule inicial que minimice la desviación a partir del mismo durante su ejecución y que la degradación del rendimiento se mantenga en un nivel aceptable. En su primer artículo, [26] los autores estudian el problema de scheduling en un entorno de single machine cuyos trabajos tienen tiempo de preparación no nulos y existen averías aleatorias de máquinas. Los tiempos de avería y de distribuciones para la duración de las reparaciones son conocidos *a priori*. En su segundo artículo [27], estudian un problema de scheduling job shop con avería de máquinas aleatoria. En ambos estudios utilizan el retraso máximo como medida de rendimiento. A diferencia de Wu et al., intentan minimizar las desviaciones mientras se genera un schedule inicial, sin reprogramar luego de una avería. La desviación del schedule inicial se mide en términos de la desviación absoluta esperada de los tiempos de completamiento de los trabajos. Se ofrece un enfoque de dos etapas; en la primera se determina una sucesión de trabajos que minimiza el retraso máximo, utilizando los algoritmos de Carlier y el algoritmo shift bottleneck para los casos de single machine y job shop, respectivamente. En la segunda etapa, inserta un tiempo inactivo en esta sucesión. Las cantidades de tiempo inactivo son suficientemente grandes para proveer una protección frente a avería de máquinas pero también son lo suficientemente pequeños para que el retraso máximo no aumente demasiado. Sus resultados computacionales indican que la estabilidad puede mejorarse fácilmente aumentando ligeramente el retraso máximo.
- O’Donovan et al., 1999, [28], combinan enfoques reactivos y proactivos y examinan la política scheduling/rescheduling usando medidas de estabilidad y de rendimiento en un entorno de single machine. El rendimiento del schedule se mide mediante la tardanza total y la estabilidad, mediante las desviaciones absolutas de los tiempos de completamiento. Existen tiempos de preparación no nulos y avería aleatoria de máquinas. Los autores utilizan ATC y ATC con inserción de tiempos inactivos para generar el schedule inicial. Las alternativas para el rescheduling son ATC (Apparent Tardiness Cost), ATC modificado (el cual calcula la holgura de un trabajo basado en su tiempo de finalización predicho y tomando en cuenta el tiempo inactivo insertado) y el scheduling right-

shift. Los resultados experimentales indican que el ATC con tiempos inactivos para el scheduling y el ATC modificado para el rescheduling son mejores, en lo que concierne a estabilidad. También consideran *trabajos sensitivos*, pues los trabajos son sensitivos a las perturbaciones que se acaban de producir y el grado de sensibilidad difiere de trabajo en trabajo. Para este modelo, los autores proponen ATC con tiempos inactivos para el scheduling y Smart ATC para el rescheduling. El Smart ATC es similar al ATC modificado pero utiliza tiempo de procesamiento afectado estimado en vez del tiempo de procesamiento de un trabajo.

- Guo y Nonaka, 1999, [9], estudian la estabilidad en un entorno flow shop con 3 máquinas. La medida de rendimiento es el makespan y la máquina intermedia está sujeta a avería de single machine. La medida de estabilidad utilizada es la primera de la sección 4.2. Los autores desarrollaron un algoritmo para reprogramar de forma óptima el sistema considerando diferentes intervalos en el tiempo de la avería de máquina (t) y la duración del tiempo de inactividad (D). Así, para diferentes valores de D y t deberían obtenerse diferentes schedules, luego de reprogramar. En este estudio proponen primero generar diversos schedules óptimos sin considerar avería de máquinas para luego analizar cada schedule óptimo de acuerdo al algoritmo propuesto para los diferentes valores de D y t , calculando así su estabilidad esperada. Finalmente seleccionan el schedule con la mínima medida de estabilidad y reprograman de acuerdo al algoritmo propuesto cuando ocurre la avería durante el proceso de implementación.
- Van der Vonder, Demeulemeester y Herroelen, 2008, [35] generan schedules iniciales estables en problemas de scheduling con recursos limitados (RCPS) con variabilidad en la duración de las actividades. Utilizan la primera medida de estabilidad dada en a sección 4.2. El enfoque utilizado es de dos etapas: En la primera etapa se genera un makespan, usando un algoritmo determinista, para después minimizarlo. La duración de las actividades se establece de acuerdo a su valor promedio. En la segunda etapa, añaden tiempos de inactividad adicionales al inicio de cada actividad para proteger el schedule contra los efectos negativos en la variación de la duración de las actividades, utilizando diversas heurísticas: VADE (Virtual Activity Duration Extension), STC (Starting Time Critically), RFDF (Resource Flow Dependent Float Factor), utilizada como benchmark y también utilizan una heurística iterativa mejorada, la cual toma un schedule generado por la heurística antes mencionada como valor de entrada. De ese modo, construyen una vecindad de la actual solución deslizando una actividad seleccionada para todos los tiempos de inicio discretos posibles dentro de una ventana de tiempo que no afecta a otras actividades. Seleccionan el mejor tiempo de inicio evaluando toda la vecindad mediante simulación y toman este schedule como valor de entrada para la próxima iteración. Finalmente proponen también una búsqueda tabú. Sus resultados computacionales muestran que la STC es, por lo general, la mejor heurística si es que no se considera una fase de mejora. También se observa que si todas las actividades están sujetas a una incertidumbre significativa, la búsqueda tabú produce los mejores resultados, aunque esta es computacionalmente costosa.
- Barber y Salido, 2015, [3] estudian la estabilidad en problemas tipo CSP al definir una s -estabilidad mediante el concepto de vecindades. Estas vecindades vienen dadas en términos de las normas definidas en el espacio n -dimensional: mientras menor sea su s -estabilidad, más estable será la solución s (schedule). Obtener la estabilidad de un CSP conlleva a un problema de optimización con satisfacción de restricciones (CSOP) y, por lo general, resolver este tipo de problemas es NP-hard. Sin embargo, este costo computacional se reduce al buscar una solución s' en la vecindad más cercana de s . Los autores desarrollaron dos algoritmos denominados: *Algoritmo 1*, (para trabajar en dominios métricos) y *Algoritmo 2* (para dominios no métricos). Ambos algoritmos utilizan un proceso llamado *estabilidad incremental*. El costo computacional para obtener la estabilidad de una solución disminuye cuando existe una solución en una vecindad cercana a s . Según este criterio, la estabilidad de una solución s representa la distancia normalizada mínima entre s y una nueva solución factible del nuevo problema luego del conjunto de incidencias.
- Glumac y Kovacic, 2019, [8], utilizan una medida de estabilidad robusta, basada en el radio espectral, para medir la robustez de los cambios de parámetros esclavos. Los autores introducen un modelo de co-simulación matricial de ecuaciones diferenciales lineal invariantes en el tiempo, utilizando métodos lineales generales. Este modelo que se utiliza permite efectuar un cálculo para obtener una medida de consistencia relativa basada en el cálculo por defecto del peor caso. Tanto la consistencia como las medidas de estabilidad se calculan basadas en el modelo lineal y pueden calcularse antes de la ejecución de la co-simulación. El radio de estabilidad de una matriz se define como el tamaño de las perturbaciones necesarias para llevar al sistema al borde de la inestabilidad. Finalmente, proponen la optimización multi-objetivo para utilizar las medidas introducidas para ajustar la co-simulación principal. Sus experimentos muestran que el error global sigue una ten-

dencia similar a la medida de coherencia propuesta. Esto sugiere que la consistencia relativa es una buena medida para indicar cuan bien una co-simulación principal se adapta al sistema acoplado. Dado que esta medida no depende de un estado inicial esto proporciona una medida para cualquier posible ejecución de un sistema lineal.

5.2. Enfoque de Planeación de Escenarios.

- Daniels y Kouvelis, 1995, [7] generan schedules iniciales robustos para afrontar la variabilidad del tiempo de procesamiento en un entorno de single machine. La medida de rendimiento utilizada es el flowtime total. Los autores proponen una representación basada en escenarios y un análisis de la incertidumbre. La medida utilizada es la segunda medida de robustez formulada en la sección 3.2.1.1. Desarrollan programas matemáticos llamados: 1) *problema de schedule robusto con desviación absoluta* (ADRSP) y 2) *problema de scheduling robusto con desviación relativa* (RDRSP). Las soluciones proporcionan schedules robustos cuando el número de escenarios es finito. Ambos problemas difieren uno del otro sólo en su definición de robustez, pues el primero mide las desviaciones a partir de un schedule óptimo absoluto, mientras que el otro las mide en forma relativa. Desarrollan un algoritmo branch and bound y dos heurísticas de relajación sustitutas de orden $O(n \log n)$ para generar una sucesión robusta. La carga computacional de este algoritmo crece rápidamente conforme aumenta el tamaño del problema. Las heurísticas precisan de un tiempo computacional mucho menor pues evalúan un menor número de permutaciones de trabajos a fin de determinar sucesiones robustas aproximadas; si se incrementa el tamaño del problema la carga computacional crece sólo moderadamente. Los autores comparan sus soluciones con las soluciones SEPT (shortest expected processing time), las cuales se usan en la práctica para generar la sucesión de trabajos óptima. En términos de robustez, las SEPT muestran un pobre rendimiento.
- Daniels y Carrillo, 1997, [6] generan schedules robustos en un entorno de single machine con incertidumbre en los tiempos de procesamiento y utilizan como medida de rendimiento el flowtime total asumiendo que el tiempo de procesamiento de un trabajo individual se especifica sólo de forma imprecisa. Esta incertidumbre es capturada en un conjunto de escenarios de tiempos de procesamiento, donde cada escenario puede realizarse con una probabilidad positiva. Se evalúan sucesiones de trabajos alternativas de acuerdo a la probabilidad de que el rendimiento del actual sistema no sea peor que un nivel objetivo dado T . El objetivo del schedule es determinar una sucesión β -robusta que maximiza esta probabilidad. Los autores desarrollan un algoritmo branch and bound para resolver este problema de programación entera así como una heurística para evaluar sólo unos cuantos escenarios de tiempos de procesamiento especialmente generados. Sus resultados computacionales indican que los schedules β -robustos funcionan muy bien con respecto a la medida flowtime total esperada y además, maximizan la probabilidad de obtener un rendimiento del flowtime que no sea peor que un cierto nivel objetivo dado.
- Yang y Yu, 2002, [41] estudian el mismo problema que Daniels y Kouvelis [7] en un entorno de single machine pero con un conjunto de escenarios de tiempos de procesamiento finitos y discretos considerando tres medidas de robustez: 1) minimizar el costo del escenario de peor caso, 2) minimizar el regret absoluto del escenario de peor caso y 3) minimizar el porcentaje de regret del escenario de peor caso. Estas tres medidas pueden obtenerse por un único método. Presentan, además un algoritmo de programación dinámica de orden $O(2^n)$, así como dos heurísticas de tiempo polinomial (heurística greedy y otra que utiliza el fujo de tiempo total promedio sobre todos los escenarios como medida de robustez sustituta). Sus experimentos computacionales indican que la heurística greedy supera a la heurística sustituta y la diferencia se vuelve más y más evidente al aumentar el número de escenarios.
- Kasperski, 2005, [17] realiza un estudio sobre robustez en un entorno de single machine con restricciones de precedencia y con un criterio de retraso máximo. Los tiempos de procesamiento de los trabajos y los plazos de vencimiento son inciertos y cada uno de ellos envuelve un rango de posibles realizaciones. Al igual que Daniels y Kouvelis, el autor considera un enfoque cuyo objetivo es minimizar el rendimiento del escenario de peor caso (minimax regret). Esta medida de robustez es la segunda definida en la sección 4.1.2. El autor desarrolla un algoritmo de tiempo polinomial, el cual usa recurrentemente el algoritmo de Lawler ([21]) en los escenarios de peor caso especialmente contruidos.
- Climent et al., 2013, [5], consideran problemas con satisfacción de restricciones (CSP) con dominios discretos y ordenados en un entorno dinámico e incierto. Los autores presentan la idea de cubrimientos, basada en la llamada “topología cebolla-[13], los cuales miden la protección de una tupla (solución) frente a las perturbaciones. Así, una tupla con más capas tiene una mayor probabilidad de permanecer válida que otra con menos capas. El algoritmo desarrollado calcula los

cubrimientos de las tuplas válidas luego de realizar una consistencia de arco global (GAC). Luego de calcular los cubrimientos, el CSP original se modela como un WCSP (modP) (CSP ponderado) asignando pesos a las tuplas válidas basado en sus cubrimientos. Este WCSP (modP) modelado se resuelve con un resolvente general WCSP. La mejor solución para modP se considera una de las soluciones más robustas para el CSP original. Los autores aplican esta técnica a unos problemas de scheduling de tipo benchmark derivados de los problemas de optimización “os-taillard-4-100-0” y “os-taillard-4-105-0”(Taillard E. [34]). Estos problemas, se componen de 4 máquinas, 4 trabajos y 4 tareas por trabajo. El modelo CSP resultante contiene 16 variables y 48 restricciones. El schedule óptimo tiene un makespan de 193s. con 7 buffers de tiempo naturales entre tareas. Con la técnica de cubrimientos, aplicada en el problema “os-taillard-4-100-0” se obtuvo un schedule con el mismo makespan pero con 5 buffers de tiempo adicionales de mayor duración, de modo que este schedule óptimo es más robusto que el anterior. Al aplicar esta técnica al problema “os-taillard-4-105-0”, el schedule resultante es aún más robusto que el anterior al poseer 13 buffers de tiempo de mayor duración. Sin embargo ya no es un schedule óptimo pues el makespan es de 198s. Los resultados experimentales indican que para el problema “os-taillard-4-100-0”, el máximo makespan permitido fue el mejor makespan conocido. Por tanto, todos los schedules factibles para este problema son óptimos. Para el problema “os-taillard-4-105-0”, el máximo makespan fue de 105 % del mejor makespan. Por tanto, los schedules obtenidos para este problema pueden y usualmente no serán, óptimos.

6. Conclusions. En este trabajo presentamos los diversos enfoques y políticas sobre medidas de robustez y estabilidad en problemas de scheduling. Revisamos, además, los estudios existentes en la literatura, los cuales consideran diversos entornos, heurísticas así como diferentes objetivos. A continuación, enumeramos las conclusiones y algunas futuras direcciones de investigación:

1. Debido a la gran cantidad de estudios realizados sobre robustez y estabilidad, es necesario tener definiciones claras y diferenciadas sobre estos dos conceptos. Las definiciones caracterizadas y formalizadas en este trabajo permiten que éstos puedan utilizarse, de forma general, en la evaluación de robustez y estabilidad para las soluciones de diversos problemas de scheduling con incertidumbre.
2. Algunos estudios existentes indican que la robustez y la estabilidad son objetivos conflictivos (e.g. Wu et al. [40]). Por lo tanto, se vuelve necesario una mayor investigación sobre la naturaleza del contraste entre estas dos métricas de rendimiento. También, es importante saber las circunstancias bajo las cuales una métrica tiene más prioridad que la otra
3. Para contrastar las diferencias entre robustez y estabilidad, las políticas y medidas de robustez y estabilidad consideradas son aplicadas a problemas de scheduling simples. Esto nos permite determinar, las diferentes formas en que una solución reacciona ante las incidencias: se mantiene, o se adapta o se mantiene durante un tiempo determinado.
4. La técnica de cubrimientos desarrollada por Climent et al. [5] nos permite obtener schedulings óptimos y robustos, con buffers de tiempo más amplios y, por tanto, capaces de absorber grandes demoras en las tareas. Sin embargo, esta técnica tiene una seria desventaja; al intentar obtener schedules más robustos se pierde la optimalidad del schedule.
5. El uso de algoritmos genéticos para resolver problemas de tipo job shop dinámicos son estudiados cada vez con mayor frecuencia. En este entorno, utilizando la definición de robustez definida por Jensen y Hansen [14] es posible concluir que existen soluciones robustas y que éstas se pueden encontrar utilizando computación evolutiva.
6. Se podrían ampliar los estudios existentes al considerar otras medidas de rendimiento, además del makespan, pues como afirman Mehta y Uszoy [26] es necesario desarrollar mejores medidas sutitutas para la estabilidad (y/o robustez).
7. La mayoría de las distintas políticas y enfoques existentes consideran las medidas de robustez y estabilidad en forma separada. Tomando en cuenta este escenario, podemos preguntarnos si es posible considerar enfoques de doble criterio, considerando a la estabilidad y la robustez en forma simultánea.
8. Como indicaciones para futuros trabajos, se pueden explorar otras medidas de rendimiento para evaluar los beneficios de considerar la robustez de una solución. Se pueden, por ejemplo, encontrar mejores reglas y medidas de rendimiento, explorar diferentes esquemas de interrupciones que caracterizan a algunos campos de aplicación particulares, o establecer ciertas medidas de robustez y/o estabilidad para problemas de scheduling difusos

7. Reconocimientos. A big part of this work was developed during the performance of the project proyecto PIBA-2-P-069-14 supported by INNOVATE PERÚ.

ORCID and License

Rafael Asmat Uceda <https://orcid.org/0000-0003-1711-7021>,
Edmundo Vergara Moreno <https://orcid.org/0000-0002-6868-7211>,
Flabio Gutiérrez Segura <https://orcid.org/0000-0003-0372-3864>.

This work is licensed under the [Creative Commons Attribution-NoComercial-ShareAlike 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Referencias

- [1] Al Fawzan, M. and Haouari M. *A bi-objective model for robust resource-constrained project scheduling*. International Journal of Production Economics, 9, 175-187, 2005
- [2] Artigues C., Billaut, J.C. and Esswein, C. *Maximization of solution flexibility for robust shop scheduling*. European Journal of Operational Research, 165, 314-328, 2005
- [3] Barber, F. and Salido, M.A. , *Robustness, stability, recoverability and reliability in constraint satisfaction problems*. Knowledge and Information Systems, 44(3), 719-734, 2015.
- [4] Chtourou, H. and Haouari, M. , *A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling*. Computers & Industrial Engineering, United Kingdom, 55, 183-194, 2008
- [5] Climent, L. Wallace, R., Salido, M. and Barber, F. , *Finding robust solutions for constraint satisfaction problem with discrete and ordered domains by coverings*. Artificial Intelligence Review, 1-26, 2015.
- [6] Daniels, R.L. and Carillo, J. E. *β -robust scheduling for single machine systems with uncertain processing times*. IIE Transactions, 29, 977-985, 1997.
- [7] Daniels, R. and Kouvelis, P. *Robust scheduling to hedge against processing time uncertainty in single-stage production*. Management Science, 41(2), 363-376, 1995.
- [8] Glumac, S. and Kovacic, Z. *Relative consistency and robust stability measures for sequential co-simulation*. Proceedings of the 13th International Modelica Conference, Regensburg, Germany, 2019.
- [9] Guo, B. and Nonaka, Y. *Rescheduling and optimization of schedules considering machine failures*. International Journal of Production Economics, 60(61), 503-513, 1999.
- [10] Goren, S. and Sabuncuoglu, I. *Robustness and stability measures for scheduling: single-machine environment*. IIE Transactions, 40(1), 66-83, 2008.
- [11] Goren, S. and Sabuncuoglu, I. *Optimization of schedule robustness and stability under random machine breakdowns and processing time variability*. IIE Transactions, 42(3), 203-220, 2009.
- [12] Herroelen, W. and Leus, R. *Project scheduling under uncertainty, survey and research potentials*. European Journal of Operational Research, N165(2), 289-306, 2005
- [13] Herrman, H. N, Scheinder, C., Moreira, A. Andrade, Jr. J. and Havlin, S. *Onion-like network topology enhances robustness against malicious attacks*. Journal of Statistical Mechanics: Theory and Experiment, 2011(1), 1-7, 2011.
- [14] Jensen, M. T. and Hansen, T. K. *Robust solutions to job shop problems*. Proceedings of the Congress of Evolutionary Computation, 1138-1144, 1999.
- [15] Jensen, M. T. *Improving robustness and flexibility of tardiness and total flow time job shops using robustness measures*. Applied Soft Computing, 1, 35-32, 2001.
- [16] Jensen, M.T. *Generating robust and flexible job shop schedules using genetic algorithms*. IEEE Transactions on Evolutionary Computation, 7(3), 275-288, 2003.
- [17] Kasperski, A. *Minimizing maximal regret in the single machine sequencing problem with maximim lateness criterion*. Operations Research Letters, 33, 431-436, 2005.
- [18] Kitano, H. *Toward a theory of biological robustness*. Mol Syst. Biol, 3(1), 137, 2007.
- [19] Kutanoglu, E. and Sabuncouglu, I. *Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop*. Journal of Manufacturing Systems, 20(4), 264-279, 2001.
- [20] Kolisch, R. , Schwindt, C. and Sprecher, A. *Benchmark instances for project scheduling problems In J. Weglarz (Ed.)*. Handbook on recent advances in project scheduling. Dordrecht: Kluwer, 1998.
- [21] Lawler, E. *Sequencing jobs to minimize total weighted completion time subject to precedence constraints*. Ann. Discrete Math, 2, 75-90, 1978.
- [22] Li, W. and Glazebrook, K.D. *On stochastic machine scheduling with general distributional assumptions*. European Journal of Operational Research, 105, 525-536, 1998.
- [23] Leung, J. Y. T. and Pinedo, M. *A note on scheduling parallel machines subject to breakdown and repair*. Naval Research Logistics, 51(1), 60-71, 2004.
- [24] Liao, F. and van Wee, B. *Accessibility measures for robustness of the transport system*. Transportation, 44, 1213-1233, 2017.
- [25] Leon, V. J. , Wu, S.D. and Storer, R. , *Robustness measures and robust scheduling for job shops*. IIE Transactions, 26(5), 32-43, 1994
- [26] Mehta, S. V. and Uzsoy, R. *Predictable scheduling of a job shop subject to breakdowns*. IEEE Transactions on Robotics and Automation, 14(3), 365-378, 1998.
- [27] Mehta, S.V. and Uzsoy, R. *Predictable scheduling of a single machine subject to breakdowns*. Int. J. Computer Integrated Manufacturing, 12(1), 15-38, 1999.
- [28] O'Donovan, R., Uzsoy, R. and McKay K. N. *Predictable scheduling of a single machine with breakdowns and sensitive jobs*. Int. J. Prod. Res., 37(18), 4217-4233, 1999.
- [29] Rossi, F., van Beek, P. and Walsh, T. *A general computational method for robustness analysis with applications to synthetic gene networks*. Bioinformatics, 25(12), 168-179, 2006.
- [30] Sabuncuoglu, I. and Goren, S. *Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research*. International Journal of Computer Integrated Manufacturing, 22(2), 138-157, 2009.
- [31] Sabuncuoglu, I. and Goren, G. *Generating Robust and Stable Schedules in a Single Machine Environment*. Industrial Engineering Research Conference, 2004.

- [32] Sevaux, M. and Sorensen, K. *A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates*. OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 2, 129-147, 2004.
- [33] Sotskov, Y. , Sotskova, N.Y. and Werner, F. *Stability of an optimal schedule in a job shop*. Omega: The International Journal of Management Science, 25(4), 397-414, 1997
- [34] Taillard, E. *Benchmark for basic scheduling problems*. Eur. J. Oper Res, 64(2), 278-285, 1993
- [35] Van der Vonder, S. , Demeulemeester, E. and Herroelen, W. *Proactive heuristic procedures for robust project scheduling: An experimental Analysis*. European Journal of Operational Research, 189(3), 723-733, 1998.
- [36] Verfaillie, G. , and Jussien, N. *Constraint solving in uncertain and dynamic environment: a survey*. Constraints 10(3), 253-281, 2005.
- [37] Wallace, R. , Grimes, D. and Freuder, E. *Solving dynamic constraint satisfaction problems by identifying stable features*. Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI-09), 621-627, 2009.
- [38] Wiggins, S. *Introduction to applied nonlinear dynamical systems and chaos*, Springer, New Yor, U.S.A., 1990.
- [39] Wu, S. D., Storer, R. N. and Chang, P. *A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness*. Operations Research, 47(1), 113-124, 1999.
- [40] Wu, D.D. , Storer, R. N. and Chang, P. *One-machine rescheduling heuristics with efficiency and stability as criteria*. Computers Ops Res., 20(1), 1-14, 1993.
- [41] Yang, J. and Yu, G. *On the robust single machine scheduling problem*. Journal of Combinatorial Optimization, 6, 17-33, 2002.