Resumen:

El Flow Shop Scheduling Problem (FSP) es un problema que busca, en la mayoría de planteamientos, reducir el tiempo de producción de un conjunto de máquinas que tienen asociados N trabajos a realizar. Una variación del FSP es el Permutation Flow Shop Scheduling Problem (PFSP) que busca encontrar, realizando permutaciones en el orden de los trabajos, reducir el tiempo requerido para completar las tareas. Por último, el objeto de trabajo del articulo se centra en la generalización del PFSP, el PFSP with Stochastic Times (PFSPST) que añade el uso de valores aleatorios a la hora de determinar el tiempo de procesado de cada trabajo.

La literatura relacionada con la variante estocástica del PFSP no es muy extensa, pues la mayor parte de artículos se han centrado en la variante determinista del problema. Por ello, algoritmo propuesto en el articulo para resolver este problema se basa en aprovechar los avances hechos en el estudio de la variante determinista para poder aplicarlos a la variante estocástica.

Concretamente, el algoritmo propuesto consta de las siguientes etapas:

1. Convertir el problema en la variante determinista (PFSP, sin tiempos aleatorios).
2. Obtener soluciones de calidad, utilizando algoritmos eficientes basados en el PFSP (e.g. Iterated Local Search).
3. Ejecutar una simulación con las soluciones obtenidas, para obtener una muestra de los “tiempos para completar los trabajos”.
4. Repetir los puntos 2-3 hasta que las condiciones de parada del algoritmo se cumplan (ya sea por tiempo o por calidad de la solución).

Valoración:

Desde mi punto de vista lo más destacable seria:

En primer lugar, la simplificación del problema de la variante estocástica en la variante determinista resulta una estrategia muy beneficiosa ya que como se afirma en el artículo, la solución del modelo determinista necesariamente estará dentro del espacio de soluciones del modelo estocástico. Además, trabajando con la variante determinista se puede sacar provecho de los mejores resultados obtenidos en otros estudios.

En segundo lugar, el uso de simulaciones para obtener los diferentes “tiempos de finalización de tareas” aporta la ventaja de no depender del uso de distribuciones estándar a la hora de determinar los tiempos de ejecución de las diferentes tareas, pues estas pueden no seguir dichas distribuciones por lo que se pueden ir adaptando a medida que el algoritmo se ejecuta.