

Resumen

Muchos de los problemas industriales se pueden modelar como un problema de scheduling donde algunos recursos son asignados a tareas a fin de minimizar el tiempo de finalización, para reducir el uso de los recursos, el tiempo de inactividad, etc. Existen varios tipos de problemas de scheduling que intentan representar diferentes situaciones que pueden aparecer en los problemas del mundo real. Job-Shop scheduling (JSP) es el problema más utilizado. En JSP hay diferentes trabajos, cada trabajo tiene diferentes tareas y estas tareas tienen que ser ejecutadas por diferentes máquinas. JSP puede ser extendido a otros problemas con el fin de simular una mayor cantidad de problemas reales. En este trabajo se ha utilizado el problema job shop scheduling con operadores $JSO(n, p)$, donde cada tarea también debe ser asistida por un operador de un conjunto limitado de ellos. Además, hemos ampliado el clásico problema JSP a un problema donde las máquinas pueden consumir diferentes cantidades de energía al procesar tareas a diferentes velocidades (JSMS). En JSMS las operaciones tienen que ser ejecutadas por una máquina que tiene la posibilidad de trabajar a diferentes velocidades. Los procesos industriales implican un gran número de problemas de scheduling de tareas. Los dos problemas extendidos pueden representar un gran número de problemas combinatorios en los procesos industriales.

Los problemas de scheduling consideran indicadores de optimización tales como: el procesamiento de tiempo, la calidad y el coste. Sin embargo, hoy en día los gobiernos y los empresarios están interesados también en el control del consumo de energía debido al aumento de la demanda y del precio de los combustibles, la reducción de las reservas de materias primas energéticas y la creciente preocupación por el calentamiento global. En esta tesis, hemos desarrollado nuevas técnicas de búsqueda metaheurística para modelar y resolver el problema JSMS. Además, se lleva a cabo un estudio comparativo para analizar el comportamiento de nuestro algoritmo contra un resolutor conocido: IBM ILOG CPLEX CP Optimizer.

La robustez es una característica común en los problemas de la vida real. Un sistema persiste si permanece en funcionamiento y mantiene sus principales características a pesar de las perturbaciones continuas, cambios o incidencias. Hemos desarrollado una técnica

para resolver el problema $JSO(n, p)$ con el objetivo de obtener soluciones robustas y optimizadas.

Hemos desarrollado un modelo dual para relacionar los criterios de optimalidad con el consumo de energía y la robustez/estabilidad en el problema JSMS. Este modelo se ha desarrollado para proteger a las tareas dinámicas contra incidencias, con el fin de obtener soluciones sólidas y que tengan en cuenta el consumo de la energía. El modelo dual propuesto ha sido evaluado con un algoritmo memético para comparar el comportamiento frente al modelo original.

En el problema JSMS hay una relación entre la eficiencia energética, la robustez y el makespan. Por lo tanto, se estudia la relación entre estos tres objetivos. Se desarrollan fórmulas analíticas para representar la relación estimada entre estos objetivos. Los resultados muestran el equilibrio entre makespan y robustez, y la relación directa entre la robustez y eficiencia energética. Para reducir el makespan, el consumo de energía tiene que ser aumentado para poder procesar las tareas más rápido. Cuando el consumo de energía es bajo, debido a que las máquinas no están trabajando a la velocidad más alta, si una incidencia aparece, la velocidad de estas máquinas puede ser aumentada con el fin de recuperar el tiempo perdido por la incidencia. Por lo tanto la robustez está directamente relacionada con el consumo de energía. Además, la robustez también está directamente relacionada con el makespan porque, cuando el makespan aumenta hay más huecos en la solución, que en caso de surgir incidencias, estas pueden ser absorbidas por estos buffers naturales.

La combinación de robustez y estabilidad da un valor añadido debido a que si una incidencia no puede ser absorbida directamente por la tarea interrumpida, esta puede ser reparada mediante la participación un pequeño número de tareas. Por lo tanto, la solución original se puede recuperar para mantener viable el schedule original. Cuando la recuperación no es posible, se debe aplicar rescheduling. En este trabajo se proponen dos técnicas diferentes para gestionar el rescheduling sobre el problema JSMS. Así que, dada una incidencia, una técnica match-up es usada para determinar el punto de tiempo donde el schedule recupera la solución original y se obtiene una solución replanificada. A continuación, se propone un algoritmo memético para buscar una solución eficiente energéticamente en la zona de replanificación establecida. Un extenso estudio se ha llevado a cabo para analizar el comportamiento de las técnicas propuestas. Para ello, se han simulado incidencias sobre algunas instancias conocidas. La técnica match-up propuesta mantiene un buen rendimiento y muchos casos han sido recuperados de una manera eficiente. Por último, la mayoría de las soluciones replanificadas han sido mejoradas para ahorrar en el consumo de energía.

Este trabajo representa un avance en el estado del arte en los problemas de scheduling y, en particular, en el problema donde el consumo de energía puede ser controlado por la velocidad de las máquinas.