

UNIVERSIDAD DE VALLADOLID

ALGORITMOS Y COMPUTACIÓN

# Job Shop Scheduling

*Sergio García Prado*

Seguimiento del trabajo en:  
[github.com/garciparedes/Job-Shop-Scheduling-NP-Complete](https://github.com/garciparedes/Job-Shop-Scheduling-NP-Complete)

December 23, 2015

# Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Reducción</b>	<b>2</b>
2.1	3-Partición . . . . .	2
2.2	Job Shop . . . . .	3
<b>3</b>	<b>Bibliografía</b>	<b>4</b>

# 1 Introducción

Job Shop Scheduling es un problema de optimización que se estudia en Ciencias de la Computación e Investigación de Operaciones. Ahora describiremos el problema y el objetivo que se pretende conseguir:

Supongamos que tenemos  $n$  trabajos a realizar, los cuales denotamos como  $J_i$  tal que  $1 \leq i \leq n$  cada uno de los cuales de distinta duración. También tenemos  $r$  máquinas con las que realizar los trabajos que denotaremos por  $M_j$  tal que  $1 \leq j \leq r$ . Cada una de estas es la encargada de realizar una tarea concreta necesaria para finalizar cada uno de los trabajos. A cada una de estas tareas las denotaremos por  $O_{ji}$  que representa la tarea que se debe realizar en la máquina  $M_j$  del trabajo  $J_i$ .

En Job Shop Scheduling el orden de las tareas necesarias para completar el  $i$ -ésimo trabajo importa y no se puede modificar.

El objetivo será tratar de minimizar al máximo posible el tiempo necesario para completar todas las tareas. Para ello trataremos el problema como de decisión, es decir, buscaremos la respuesta la siguiente pregunta:

**¿Se pueden resolver los  $i$  trabajos a partir de las  $j$  máquinas en un tiempo menor o igual que  $r$ ?**

## 2 Reducción

Para demostrar que el problema Job Shop Scheduling pertenece a la clase NP-Complete nos basaremos en unos problemas cuya pertenencia a dicha clase ya está demostrada, por lo cual nos bastará demostrar que JobShop se puede reducir a alguno de estos:

### 2.1 3-Partición

**3-Partición** es un problema NP-completo, que consiste en decidir si dado un multiconjunto (conjunto en el cual cada miembro del mismo tiene asociada una multiplicidad)  $S$  de  $n = 3m$  enteros positivos, puede ser particionado en  $m$  subconjuntos  $S_1, S_2, \dots, S_m$  tal que la suma de sus elementos sea la misma.

Lo interesante del problema 3-Partición es que sigue siendo NP-completo incluso cuando los enteros de  $S$  están acotados por un polinomio en  $n$ . En otras palabras, el problema permanece al conjunto de problemas NP-completo incluso cuando la representación de los números en el argumento de entrada es unaria (representación mediante la repetición de un único símbolo). Es decir, el problema de la 3-partición es NP-completo en sentido estricto o estrictamente NP-completo. Esta propiedad, y en la 3-partición en general, es útil en muchas simplificaciones donde los números están representados de forma natural en unario.

Este problema fue demostrado por primera vez por Garey y Johnson mediante una solución de búsqueda 3-dimensional.

## **2.2 Job Shop**

### 3 Bibliografía

-