Production and Transportation Planning Problem Instance Database Deliverable D28, ALCOM-FT Project

Peter Lennartz

Institute of Information and Computing Sciences
Utrecht University
Padualaan 14, 3584 CH Utrecht
The Netherlands
peterl@cs.uu.nl

Abstract. The job shop scheduling problem (JSP) is considered to be one of the hardest (and thus one of the most interesting) combinatorial problems by many researchers. Plenty of algorithms have been developed in the last few years and there was a need of a database of test instances to compare these algorithms (like TSPlib for the TSP). Within this deliverable we propose some instances of 'real life' production and transportation planning for this database. These instances come from the pharmaceutical industry, and they possess some specific characteristics, like the so called no-wait constraints.

1 Introduction

The no-wait job shop scheduling problem NWJSP is a special case of the job shop scheduling problem, which can be defined as follows. Given a set of tasks (called the operations) each task is to be executed on a machine of a certain machine type, which takes a given amount of time (the processing time); there are several identical machines of the same type available, and each machine can handle at most one operation at a time and is continuously available from time zero onwards. Furthermore we are given a set of so-called no-wait constraints, where each constraint decrees that a given operation has to be started *exactly* Δ time units after another, given operation has finished; this number Δ can be negative, which implies that the operations overlap in their execution. The objective is now to find the shortest schedule; that is an assignment of starting times to the operations such that the no-wait constraints and the machine capacities are fulfilled (i.e., the schedule is valid) and that the point in time at which the last operation finishes processing is as small as possible. The no-wait constraints are added to model the need to process an intermediate product immediately, because it is unstable.

The purpose of this deliverable was to gather some 'real life' instances of this type of problem and to make them accessible for research purposes.

2 Problem instances

The instances within this deliverable are between 33 and 46 jobs in size. It is intended to be an ultimate test for exact algorithms. There already exist heuristics that give quite reasonable answers for instances of this size.

Now we have come to the format of the data, and after we have explained this we will describe how to obtain these instances from the problem instance database.

2.1 Format

The data comes in standard ASCII text files that follow the DOS line feed conventions (i.e., $0x0d\ 0x0a$ instead of $0x0d$ on UNIX systems). When we open one of these text files with an editor we see that each of them consists of a list of integers. These integers are to be interpreted in the following way: The first integer is the number of machine types. It follows a list of machine identifiers and numbers of machines of this specific type.

The next integer is the number of jobs. It follows a list consisting of job IDs and the description of the jobs. Each job description consists of the following data. First, the number of operations, which is followed (within the job description) by a list of operation descriptions. This list contains (in this order) the operation identification number, the required time between the starting time of this operation and the starting time of the first operation of the job, its processing time, and the resource type number on which it has to be executed.

2.2 Availability

Because these instances come from an actual industrial company and might give some insight in their internal production circumstances we cannot publish the described material on the web. If you are interested in this material you can instead write an e-mail to Peter Lennartz (peterl@cs.uu.nl). A copy of the data will be sent to you.