Department of Mathematics and Computer Science
University of Southern Denmark, Odense

March 6, 2008
Marco Chiarandini

DMP87 - Scheduling, Timetabling and Routing
Weekly Notes
Lecture 12, Fall 2008

Lecture March 6

We treated the following:

- Tabu search for $Jm \mid \mid C_{max}$ (slides or photocopies for lec-12.pdf available from the Documents section of the Black Board)
- Job shop generalizations (photocopies lec-12.pdf available from the Documents section of the Black Board).
- Resource Constrained Project Scheduling, Definitions (Slides)
- RCPSP model applications (chalk and board)

The next week there are no lectures. It is a good chance to repeat the material treated so far and prepare for the launch of the project, which is foreseen for week 13-14. It is recommendable to prepare for the project making acquaintance with the general solution methods.
The next lecture is scheduled for Monday, March 17. We will treat the following:

- Heuristics for RCPSP (photocopies lec-13.pdf from the Documents section of the Black Board).
- Timetabling, Reservations (textbook pages 205-221)
- Discussion on the assignments below

Assignment 1

Design one or more schedule generation scheme for the RCPSP, that is, a procedure that takes an ordered list of activities and outputs the starting times of the activities such that the resulting schedule is feasible.

Assignment 2

Using job shop models and generalization thereof model the following cases.

Robotic cells [1]

A *robotic cell* is a workshop consisting of m machines, in which one *robot* is in charge of moving the parts from each machine to the next, and between the machines and the input/output stations. In particular, in a *robot-centered cell*, the machines, the input and the output devices are arranged along a semi-circle, and the robot arm rotates around the center of this semi-circle, thus reaching all the machines (see Figure 1). In a *mobile robot cell*, the robot can move (e.g. by means of rails) within the cell to perform its activities at different locations. In

an *in-line robot cell*, parts move along a linear conveyor and at any time the m robots perform their activities in parallel. In a loop cell, the transfer device is actually an automated guided vehicle (AGV) that unidirectionally moves among the cells and a unique input/output device.

We are interested in finding the optimal schedule of parts and robot moves with respect to some criterion. A schedule is a feasible assignment of starting times to each operation of each part, and it can be seen as the combination of two different decisions: (i) the order in which different parts enter the system, (ii) the sequencing of robot moves.

The machines have no input or output buffering facilities, each machine can process one part at a time, and a part is allowed to wait on the machines for the robot to pick it up. After processing a part, a machine is blocked until the robot comes and removes the part. This case is referred to as *bufferless robotic cells*. Alternatively, the parts must satisfy the no-wait constraint, i.e., as soon as a part is completed by a machine, the robot must immediately move it to the next machine (or to the output station). In other words, machines are not allowed to act as buffers. We refer to this case as *no-wait robotic cells*. No-wait processing is typical of the steel industry, in which the machines are indeed large treatment furnaces, the parts are steel ladles and the robot is a crane in charge of moving the ladles around the shop.

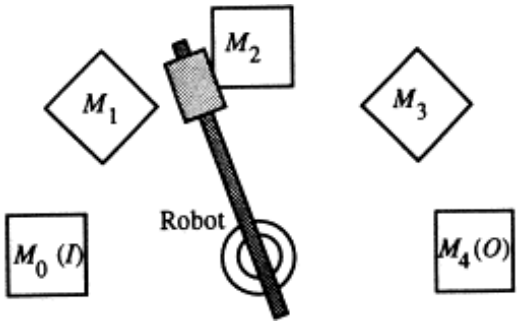

Figure 1: The layout of a robotic cell.

The goal is to find a schedule that maximizes the throughput.

Model this problem and construct the alternative or disjunctive graph to represent the small case Figure 1.

Train dispatching [2]

Railway operators plan train services in detail, defining several months in advance the train order and timing at crossings, junctions and platform tracks. A robust timetable is able to deal with minor delays occurring in real-time. However, technical failures and disturbances may influence the running times, dwelling and departing events, thus causing primary delays. Due to the interaction between trains, these delays may be propagated as secondary delays to other trains in the network. A partial modification of the timetable may then be required during operations. Hence, managing railway traffic in real-time requires re-scheduling train movements through the network, minimizing secondary delays and ensuring the feasibility of the resulting plan of operations. This process requires effective solutions within minutes and is called train dispatching, or conflict resolution.

In its basic form a railway network is composed of track segments and signals. Signals allow control of traffic on the network, and the avoidance of any potential collision between trains. There are block signals before every junction as well as along the lines and inside the stations. A block section is a track segment between two block signals.

A signal aspect may be either red, yellow or green. The red aspect means that the subsequent block section is either out of service or occupied by another train, the yellow aspect means that the subsequent block section is empty, but the one after is occupied by another train, and a green aspect means that the two subsequent block sections are empty. A train is allowed to enter the next block section at high speed only if the signal is green, it has usually to slow down to a maximum speed of 40 km/h if the signal aspect is yellow, and must stop before the end of the current section if the signal aspect is red. Thus, each block section can host at most one train at a time.

A block section takes a given time to be traversed which is known in advance for each train since all trains travel at their programmed speed whenever possible. A delay may occur at the end of a block section if the signal aspect is red. The traversing time of the subsequent train starts when the head (the first axle) of the train enters the new block section. However, the previous block section remains blocked by the train for a certain amount of time. This takes into account the time between the entrance of the head of the train in a block section and the exit of its tail (the last axle) from the previous one, plus an additional safety margin. After this time the signal before the previous section is switched to yellow and one before that to green.

A conflict occurs whenever two or more trains require the same block section at the same time. The real-time conflict resolution problem (CRP) is the on-line problem of finding a conflict-free schedule compatible with the real-time status of the network and such that trains arrive and depart with the smallest possible delay. In other

words, solving CRP requires assigning starting times $t_1, \ldots, t_n$, i.e., defining the exact time each train can enter each block section.

Figure 2 illustrates a small conflict resolution problem with two trains traveling at different speed. The railway network has four block sections (denoted as 1, 2, 3 and 9), a simple station with two platforms (6 and 7), and three junctions (4, 5 and 8). All these resources have a capacity of one. At time $t = 0$ there are two trains in the network. Train T_A is a slow train running from block section 3 to block section 9, and stopping at platform 6. Train T_A can therefore enter a block section only if the signal aspect is yellow or green. Train T_B is a fast train running from block section 1 to block section 9 through platform 7 without stopping. Train T_B can therefore enter a block section at high speed only if the signal aspect is green.

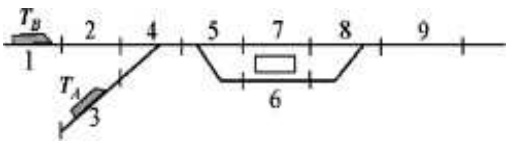

Figure 2: A small rail network.

Model this problem and construct the alternative or disjunctive graph to represent the small case in Figure 2.

References

[1] A. Agnetis. Scheduling no-wait robotic cells with two and three machines. *European Journal of Operational Research*, 123(2):303–314, 2000.

[2] Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.